UNIVERSITY of L'AQUILA

Department of Engineering

Master Thesis
in
Mathematical Modelling in Engineering

Finding Minimum Energy Paths on Error Affected Potential Energy Surfaces

Advisor
Professor Leonardo Guidoni

Co-Advisor
Doctor Emanuele Coccia

Candidate
Miriam García Soto

Mathmods Master Program

With the support of the Erasmus Mundus program of the European Union

Academic Year 2011-2012

Abstract

Chemical reactions represents fundamental processes through which the chemical composition of molecules and solids is transformed. During reactions chemical bonds between atoms are broken or formed passing from an initial stable state, identified as the reactants, to a final stable state, identified as the products. During this transformation the atomic configuration is changing following a pathway which can be identified by a collective reaction coordinate. The identification of such pathway is of key importance to understand the reaction mechanisms and to determine the kinetic properties, i.e. the velocity of the considered reaction.

Through molecular simulations is possible to assign to each three-dimensional molecular configuration $\vec{R} \in \mathbb{R}^{3N}$ of a molecule with N atoms a potential energy surface represented by $\vec{R} \times V(\vec{R})$. This function V describes the potential energy of the system and locates the stable molecular configurations, as reactant and product states, in a local minimum. Using minimization algorithms one can optimize an initial geometrical conformation reaching one of these minima and therefore obtaining information about the molecular geometries for the stable molecules. The search for pathways connecting different minima requires special algorithms to be achieved.

From a geometrical perspective this problem corresponds to draw a pathway in the $(3N + 1)$-dimensional euclidean space, lying on the potential energy surface, connecting the atomic configurations corresponding to two different minima (reactants and products) of this potential energy surface and passing through one ore more intermediate saddle points. The reaction pathway can be defined as the Minimum Energy Path (MEP), which is the energy path between the initial and final minima constructed by the union of steepest descent paths from the intermediate lowest saddle points on the energy surface to these minima. There exist several methods to find the minimum energy paths and, consequently, the saddle points on a given energy hypersurface. In this work we will consider and describe with details the Nudged Elastic Band (NEB) method.

In this thesis we will implement the NEB method in a general computational framework so that it can be easily adopted to different molecular models, from classical to quantum simulations. In the second part we will investigate, for the first time, the behaviour and performances of the NEB algorithm in energy surfaces that are affected by stochastic errors. This case is important since it simulates the application to Quantum Monte Carlo methods, an emerging techniques in quantum chemistry that provides energies and forces with a statistical error. Our results demonstrate that the introduction of the statistical error in a realistic regime does not alter significantly the overall performance of the NEB algorithm.

Contents

List of Figures

List of Tables

Acknowledgments

I thank Pr. Leonardo Guidoni for the opportunities and for his nice way of leading, and Dr. Emanuele Coccia for his generosity and all his explanations.

I am also grateful to my parents because of their support, and to Jesús for understanding and encouraging me.

Chapter 1

Molecular Simulations

Different levels of theory can be used to study the properties of molecular systems. The different approach differs by the approximations which are made in describing the system and in the description of the molecular forces. In Quantum Mechanics (QM) approaches electrons and nuclei are both explicitly considered, and the corresponding Schröndigner equation is the fundamental equation to be solved, while in Molecular Mechanics (MM) the electronic detail is not explicitly considered. The nuclei are therefore moving in an empirical potential which is function of their position only.

In this chapter we shortly review the methods used to treat molecular systems. The calculations of molecular potential energy surfaces are indeed the core of the algorithms to search for transition states which are the subject of our work.

1.1 Quantum Mechanics

In molecular systems is defined the Hamiltonian operator, $\hat{H}$, which represents the total energy of the wave function:

$$\hat{H} = -\sum_{A} \frac{1}{2M_A}\nabla_A^2 - \sum_{i} \frac{1}{2}\nabla_i^2 + \sum_{A>B} \frac{Z_A Z_B}{R_{AB}} - \sum_{A_i} \frac{Z_A}{r_{A_i}} + \sum_{i>j} \frac{1}{r_{ij}} \tag{1.1}$$

where the terms express, in this order, the kinetic energy for the nuclei, the kinetic energy for the electrons, the repulsion among nuclei, the attraction electrons-nuclei and the repulsion among the electrons. The first two terms are the part of the operator for the kinetic energy while the other three define the potential energy operator. Capital letters as A,B refer to nuclei, integers as i,j to electrons and Z represents the nuclear charge. It is considered that electron charge and mass have value 1, while M_A refers to the nuclear masses. R is representing nuclear coordinates and r electronic coordinates.

The Hamiltonian is a second order differential operator, due to $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ being x,y and z the Cartesian coordinates used to determine the positions.

The **time-dependent Schröndinger equation** defines how the quantum state of a physical system varies in time[1]:

$$i\hbar\frac{\partial}{\partial t}\psi = \hat{H}\psi \tag{1.2}$$

where

- ψ it is the wave function defining the quantum state of the physical system. It is a function with two inputs, $\psi(\vec{R}, t)$:
  . $\vec{R} \in \mathbb{R}^{3N}$ is the position of the N quantum particles (electrons and nuclei for instance) of the system.
  . t $\in \mathbb{R}$ is referring to time.
- $\hbar$ is known as the reduced Planck's constant, which is the Planck's constant ($h = 6.62606896 \times 10^{-34} m^2 kg/s$) divided by 2π.

If one is interested only in stationrary states of the molecular hamiltonian, the **time-independent Scröndinger equation** can be used. It is a partial differential eigenvalue function

$$E\psi = \hat{H}\psi \tag{1.3}$$

being E the energy of the state. The energy E_n gives the eigenvalues and for each allowed value, the Hamiltonian operator, $\hat{H}$ acts on an eigenfunction, ψ_{E_n}

$$E_n\psi_{E_n} = \hat{H}\psi_{E_n} \tag{1.4}$$

The Schröndinger equation can be solved analytically for only a few simple problems, imposing some requirements as boundary conditions and that the wave function multiplied by its complex conjugate is the normalized probability density of having the particles on position $\vec{R}$

$$\int_{-\infty}^{\infty} \psi^*\psi = \int_{-\infty}^{\infty} |\psi|^2 = 1 \tag{1.5}$$

Since we are only able to solve for some specific and easy problems, there is an assumption which simplifies the wave function. Considering that the nuclei masses are much bigger (about 2000 times) than the electrons masses it is possible to split the wave function as a product of two functions:

$$\psi_{total} = \psi_{electrons}\psi_{nuclei} \tag{1.6}$$

thus decoupling nuclear and electronic motion.

This is called the Born-Oppenheimer approximation. When it is used, the nuclei are considered as parameters and the equation is solved for the electrons at fixed nuclear positions.

1.2 Quantum methods affected by stochastic error

Different techniques have been developed during the last 50 years by physicists and chemists to solve numerically the equation 1.1 for molecules and solids. Approximations of different kind are necessaries to approach systems composed of several atoms and tens or hundreds of electrons. An overview of these methods can be found in [2]. All these methods within their approximations, can provide an evaluation of the total energy and the forces of a common atomic configuration with a precision that can be easily reduced as small as desired with an affordable computational cost. Recently Quantum Monte Carlo (QMC) methods [3, 4, 5, 6, 7, 8, 9] are emerging technique to solve electronic many-body problems.

The Variational Monte Carlo (VMC) consists in the stochastic integration of the expectation value of the Hamiltonian on a given ansatz wave function.

Correlated many-body wave functions used in VMC have a given functional form, determined by a finite (large) number of variational parameters usually obtained through an optimization procedure, with a statistical iterative technique that is converging to the lowest variational many-body wave function for the system. One advantage of this approach is that wave function parametrizations that go beyond the usual expansion in Slater determinants can be often implemented in a simple way and without significant computational overload. In particular, electronic correlation can be described through the Jastrow factor, a bosonic term (positive and symmetric under electron permutations) depending explicitly on the electronic and nuclear positions. Another group of QMC methods is based on the stochastic solution of the Schrödinger equation through projection techniques. For instance in the Diffusion Monte Carlo (DMC) technique, the ground state component of a given trial function is extracted by a long enough imaginary time diffusion process [4, 5, 8].

QMC methods have been successfully applied to study different systems such as: materials [5, 10, 11, 12], hydrogen bonding [13] and Van der Waals [14] networks and electronic excitations in gas phase [15, 16, 9] and within a QM/MM approach.

One characteristic of QMC methods is that all the calculated quantities are affected by stochastic errors. Since these errors are slowly decreasing as a function, $1/\sqrt{T}$, of the computational time T, the applications of standard tools for reaction pathway search has to be adapted to this characteristic.

1.3 Molecular Mechanics

From the point of view of the Molecular Mechanics the electronic motion is ignored and the total energy V is a function of the nuclear positions [1]

$$V(R) = \sum_{i \in bonds} \frac{k_i}{2}(l_i - l_i^0)^2 + \sum_{i \in angles} \frac{k_i}{2}\left(\theta_i - \theta_i^0\right)^2 + \sum_{n \in torsions} \frac{V_n}{2}\left(1 + \cos\left(n\omega - r\right)\right) + \\ + \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\phi\varepsilon_0 r_{ij}}\right) \quad (1.7)$$

The first term models the interaction between bonded atoms as a classical harmonic potential (a in figure 1.1), where l_i is the bond length and l_i^0 the reference bond length. k_i is the harmonic constant, which can be different for each bond.

The second term represents the contribution given by bond angles (b in figure 1.1); an harmonic potential which models the relation between groups of three bonded atoms, θ_i is the angle they form and θ_i^0 the reference angle. As in the case of bond stretching k_i is the harmonic constant, in this case, for each angle.

The third term is a torsional potential modelling how the energy changes as a function of a dihedral angle (i.e., the interaction between 4 atoms as it is appearing in the c design of figure 1.1). To define a dihedral angle four bonded atoms are selected, the first three form a plane and the last three form a second plane, and the angle determined by these two planes is the dihedral one, ω. V_n is related to the bond rotation barrier, n is called multiplicity and is the number of minima in the expression varying the dihedral angle from 0 to 2π; and γ is the phase factor.

The fourth term includes the contribution of the non-bonded interactions (d in figure 1.1). The Coulomb potential is used for electrostatic interactions and is obtained by adding all the interactions between point charges. The Lennard-Jones potential involves van der Waal forces, and describes repulsive steric effects between atoms not covalently bonded.

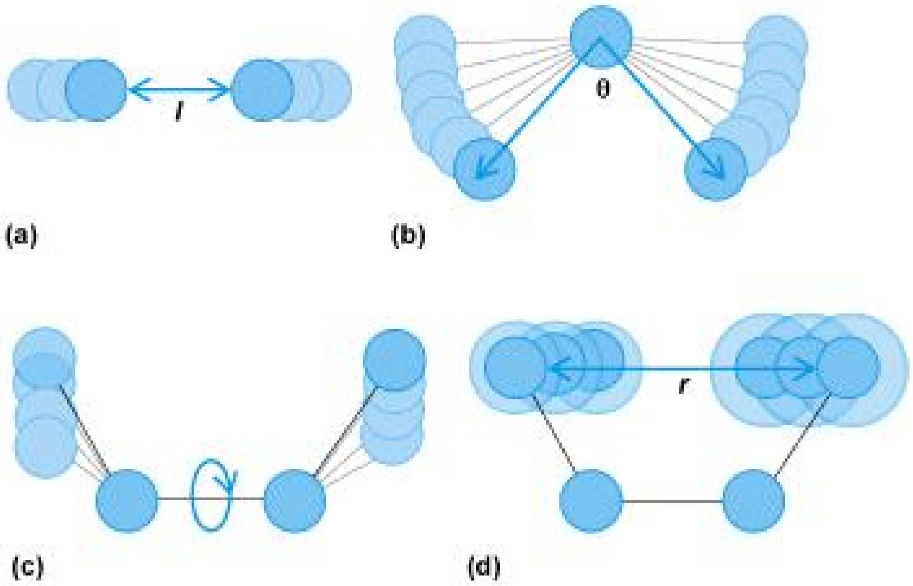


Figure 1.1: Graph showing all the force field interactions affecting the atoms, (a) the bond stretching, (b) the angle bending, (c) the torsion and (d) the non bonded interaction.

http://accessscience.com

From a computational point of view, using Molecular Mechanics allows to study systems with 10^6 atoms, thanks to the simplified description of the interactions; calculations based on the Schröndinger equations are much more accurate but so far limited to few hundreds of atoms.

Chapter 2

Potential Energy Surface and Minimization techniques

Once that has been posed the way to get the energy of a system, from a Quantum Mechanics (QM) and/or Molecular Mechanics (MM) perspective, it is possible to introduce the concept of Potential Energy Surface (PES) as the mathematical or graphical relationship between the energy of a system and its geometry[17].

2.1 Coordinates

Several different types of coordinates are used to define a system with N particles. One kind of coordinates is given by the Cartesian coordinates, (x_1, x_2, x_3) for each atom, having in total $3N$ coordinates. Other common possibility is to pose $3N - 6$ internal coordinates. The internal coordinates refer to the bonds, angles and torsional angles determining the system, terms that are mentioned in section 1.3.

The coordinates fully determine the geometry of the system, knowing one of the coordinates type it is possible to get the other one by a simple geometric transformation.

2.2 Energy

The energy of a system is given by the position of each particle, $V(\vec{R})$, so it is a multidimensional function of the coordinates, which represents the PES. It is important to know that there may be a very large number of minima on the energy surface, knowing the one with the lowest energy as the global energy minimum and the points with lowest energy value in an open interval contained in the initial domain as local minima . All these minima, from a chemical point of view, are referring to equilibrium states of a system. To identify those minimum points it will be used a minimization algorithm. There exist several methods but we will focus on the approaches most common in molecular modelling.

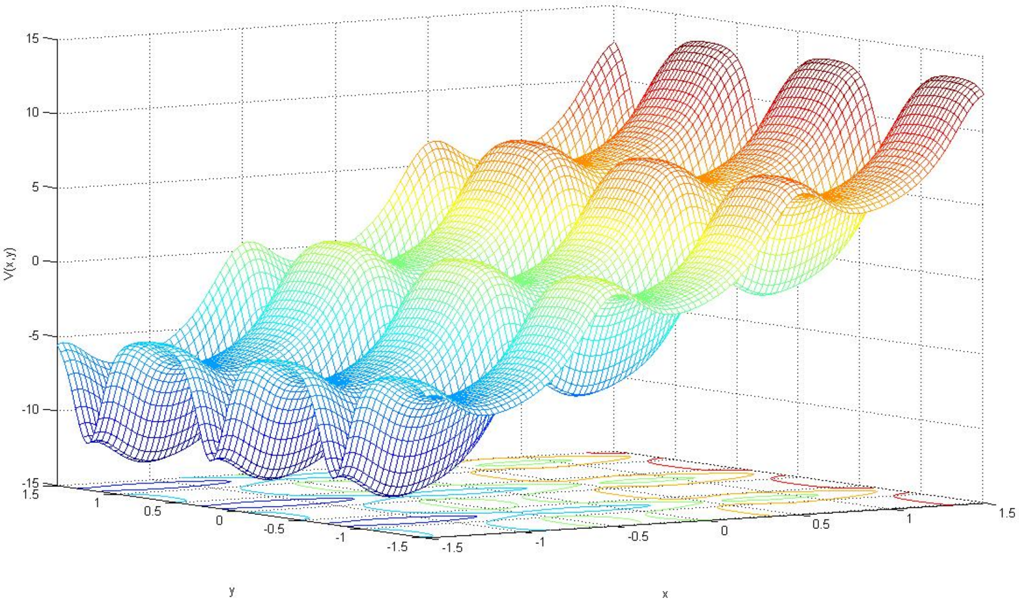


Figure 2.1: Example of a Potential Energy Surface (PES) containing several local minima.

It is also interesting to study the conformational changes between two minimum energy structures, in order to know the variation of the relative position of the atoms during a chemical reaction or a diffusion event. The highest point between two minima is a really special one because is a saddle point. Both minima and saddle points are stationary points on the energy surface, and its first derivatives respect to all the coordinates are zero. The importance of well describing the saddle point will be pointed out in the next chapter.

2.3 Energy Optimization Methods

The minimization problem can be formulated as the following:

Being V an energy function depending on the positions $R_1, R_2, ..., R_i, ...,$ find the values of those variables such that $V(R_1, R_2, ..., R_i, ...) = V(\vec{R})$ is a minimum value.

$$\min_{\vec{R}} V(\vec{R}) \tag{2.1}$$

At a minimum point the first derivative of the function respect to every variable is zero, while the second derivative is positive:

$$\frac{\partial V}{\partial R_i} = 0; \qquad \frac{\partial^2 V}{\partial R_i^2} > 0$$

The choice of the computationally best algorithm depends on the balance between the velocity on getting the solution and the quantity of memory used, the goal is to get the fastest method using the least amount of memory. The best methods are different for Quantum Mechanics and Molecular Mechanics. One of the main differences between minimization algorithms is the use of derivatives or not, and different categories of methods will be reported next. The use of second derivatives gives, in principle, the most accurate results but methods exploiting the Hessian matrix cannot be applied to large systems, since they become dramatically slow and memory demanding.

The **general scheme** for a minimization method is:

```
V is the function
$\vec{R}_j$ is initial position
while $R_j$ is not minimum (different criteria for each method)
    set new position $\vec{R}_{j+1} = f(\vec{R}_j)$
return $\vec{R}_j$
```

Non-derivative Optimization Methods

They do not involve any derivative of the function in the minimum search.

The Simplex method

A simplex is a polytope with n+1 vertices, being n the dimension of the space where the energy is computed. For instance, an energy function defined on x and y coordinates will define a simplex as a triangle.

The simplex algorithm locates a minimum by performing successive pivot operations which give an improved basic feasible solution; the choice of the operation at each step is largely determined by the requirement that this pivot does improve the solution. The possible operations are:

- reflection of the vertex with the highest value through the opposite face of the simplex;
- reflection and expansion;
- contract the simplex along one dimension, from the highest point;
- contract the simplex in all directions.

Usually the simplex method is not considered good for quantum mechanics calculations because it can be quite expensive in terms of computing time.

The Sequential Univariate Method

This method is better for Quantum Mechanics calculations, and also easier than the simplex algorithm to implement. The idea behind is to assume that the function to minimize is parabolic in all the dimensions and obtain the minimum for this parabolas in each coordinate.

- Choose step h
- ∀ i coordinate index
  Compute $R_i^* = R_i + h$ and $R_i^{**} = R_i + 2h$
  Compute R_{min} in the parabola passing through R_i, R_i^* and R_i^{**}
  Substitute R_i by R_{min}
- Repeat till the changes in all the coordinates are sufficiently small and convergence is achieved.

Derivative Optimization Methods

This family of methods uses first derivatives of the function to minimize. In the case of energy, such approach provides really useful information, the force acting on the system is the minus gradient of the potential function V

$$\vec{F} = -\frac{\partial \nabla V}{\partial \vec{R}}, \tag{2.2}$$

which is pointing to the minimum energy configuration. The second derivatives

$$\frac{\partial \vec{F}}{\partial \vec{R}} \tag{2.3}$$

indicate the curvature of the function, which gives information on where the function is changing direction, meaning by direction uphill or downhill.

First-order Methods

Steepest Descent

The Steepest Descent (SD) is a popular method in Molecular Mechanics when the initial chosen point is far from the minimum. Given an initial configuration $\vec{R}_j$ and computing the force vector in that position, $\vec{F}_j$, the iteration step is

$$\vec{R}_{j+1} = \vec{R}_j + h\vec{F}_j \tag{2.4}$$

being h an adjustable parameter.

After some iterations the point with force close to zero it will be reached; only in the case of having flat regions this algorithm will be not able to achieve the minimum.

This will be the minimization method used in this work, as it will be shown in the next chapters.

Conjugate gradients

The Conjugate Gradient (CG) method improves upon the SD method by following the conjugate search directions instead of the force. It is a common method in Molecular Mechanics when the initial position is close to the minimum.

The main steps are:

- Initialize the search direction along the force
  $\vec{d}_0 = \vec{F}_0$
- Compute the step size λ $\vec{R}_{j+1} = \vec{R}_j + \lambda\vec{d}_j$
- Set the new conjugate search direction $\vec{d}_{j+1} = \vec{F}_{j+1} + \gamma\vec{d}_j$
  with $\gamma = \dfrac{\vec{F}_{j+1}(\vec{F}_{j+1} - \vec{F}_j)}{|\vec{F}_j|^2}$

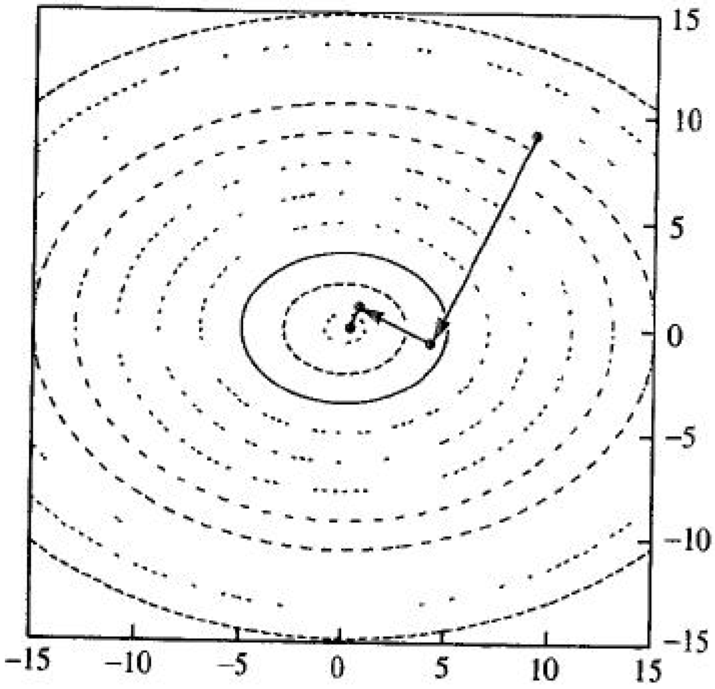


Figure 2.2: Application of steepest descents to the function $x^2 + 2y^2$

Molecular Modelling Principles and Applications; A.R.Leach (2001)

Second-order Methods

The second order methods build up information about the second derivatives and use them to step toward the minimum. The most used second order methods are the Quasi-Newton Methods, usually applied in Quantum Mechanics. Newton's method assumes that the function can be locally approximated as a quadratic one in the region around the minimum, and uses the first and second derivatives to find the stationary point. They do not compute directly the Hessian matrix but approximate by analysing successive gradient vectors.

Limited-memory Broyden-Fletcher-Goldfarb-Shanno

In this method the inverse of the Hessian matrix H^{-1}, where

$$H = \begin{pmatrix} \dfrac{\partial^2 V}{\partial R_1 R_1} & \dfrac{\partial^2 V}{\partial R_1 R_2} & \cdots & \dfrac{\partial^2 V}{\partial R_1 R_N} \\ \\ \dfrac{\partial^2 V}{\partial R_2 R_1} & \dfrac{\partial^2 V}{\partial R_2 R_2} & \cdots & \dfrac{\partial^2 V}{\partial R_2 R_N} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 V}{\partial R_N R_1} & \dfrac{\partial^2 V}{\partial R_N R_2} & \cdots & \dfrac{\partial^2 V}{\partial R_N R_N} \end{pmatrix} \tag{2.5}$$

is constructed iteratively, starting from a diagonal matrix. Two options can be considered:

Search direction $\vec{R}_{j+1} = \vec{R}_j + \vec{F}_j H_j^{-1}$ $\vec{d}_j = \vec{F}_j H_j^{-1}$ Compute the step size λ $\vec{R}_{j+1} = \vec{R}_j + \lambda \vec{d}_j$	Use directly H^{-1} $\vec{R}_{j+1} = \vec{R}_j + \vec{F}_j H_j^{-1}$

Chapter 3

Minimum Energy Paths

3.1 Energy barrier

The interest of the thesis is centred on the energy profile of chemical processes, related to the reaction rate, which means how fast the reaction takes place, from one chemical structure in equilibrium to another structure in equilibrium (two distinct minima in the potential energy surface). To understand the kinetics it is necessary to investigate the nature of the energy surface away from the minimum points. It is essential to understand what geometrical changes are involved on the kinetics and how the energy varies during the transition (figure 3.2). The initial minimum point is known as reactant, the final one as product and the energy path between them as reaction path. In this context the reaction is going to be considered a variation on the internal coordinates of the system (bonds, angles, dihedral angles); the atomic configuration of the system is changing. For a chemical reaction to occur there is an energy value which must be overcome, the activation energy. It is the minimum energy required to activate the chemical process and it is denoted as E_a. The energy barrier is the maximum value of the energy along the transition pathway, and the configuration of the system with this energy value is called the transition state.

When a system moves from one minimum configuration to another minimum configuration it is quite intuitive to think the energy increases from the initial minimum (reactant) to a maximum point in the reaction pathway and then falls to the final minimum configuration (product). The maximum is a first-order saddle point, so the derivatives of the energy function on it are equal to zero and the Hessian matrix has just one negative eigenvalue, what means it is a minimum configuration for every direction perpendicular to the energy transition pathway but not for the direction along the path, where it is a maximum. Since a chemical system tends to adopt the lowest energy configuration, it is reasonable that even during a chemical reaction is occurring and the system configuration moves away from the equilibrium state, this system evolution will use the least possible quantity of energy. The saddle point corresponds to the transition state and marks the energy barrier.

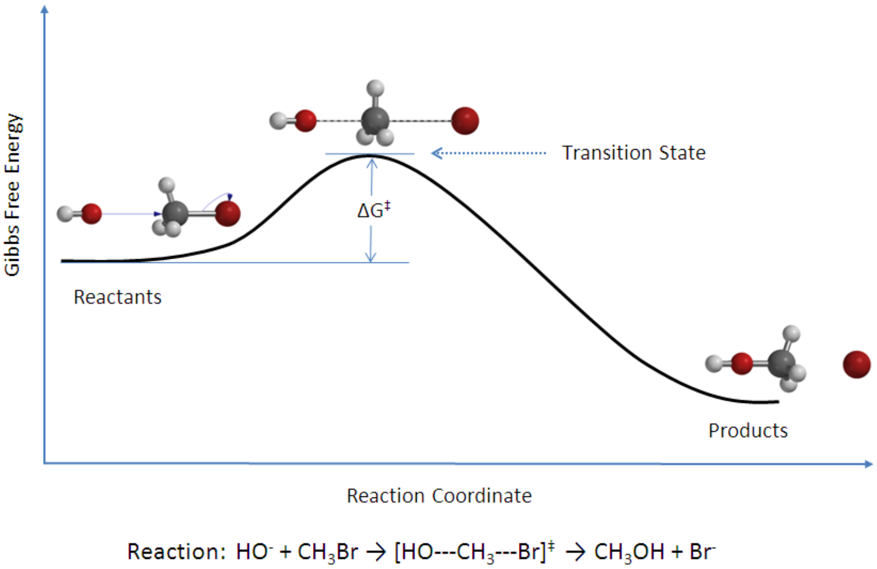


Figure 3.1: Plot of a reaction path where the configuration change (destruction and creation of bonds) of a chemical system is represented by the evolution of the reaction coordinate and the energy values related.

$$E_a = \Delta G^\ddagger$$

Understanding completely the mechanism and the energy characterization of chemical reactions is an important and challenging task on theoretical chemistry. Now, once the thesis motivation is posed, it is necessary to define and explain all the concepts and tools useful to achieve that purpose.

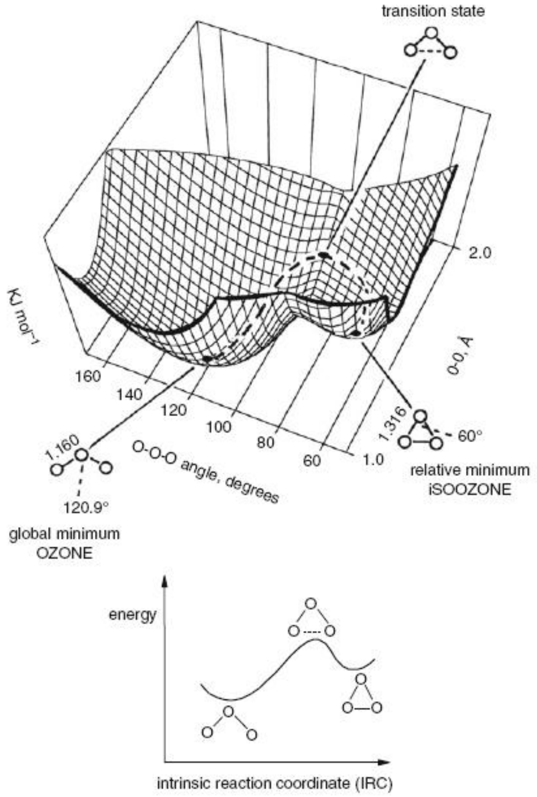


Figure 3.2: Upper bi-dimensional energy surface with a dashed line as the reaction path going from a global minimum to a relative one, which represents different configurations for the same quantity of atoms; lower the plot of the reaction path sketching the relationship between the reaction coordinate and the energy of the system

Computational Chemistry; E.G. Lewars; Kluwer Academic Publishers (2003)

3.2 Transition State Theory

The interaction energy among nuclei and electrons can be obtained from an approximate solution of the Schröndinger equation (section 1.1) or from an otherwise determined potential energy function, as the one explained in section 1.3, in which the electronic detail is neglected. For obtaining accurate estimates of reaction the Transition State Theory (TST) is applied, that uses purely statistical methods. The TST depends on, mainly, two assumptions[18]:

- A Boltzmann distribution is established and maintained on all the transition configurations (reactant states)
- The reacting trajectory, going from the product to the reactant crosses a dividing surface of dimension d-1, when d is the degrees of freedom in the system, only once.

The rate constant is defined as

$$k_{reactant \to product} = Ae^{-\frac{E_a}{k_B T}} \tag{3.1}$$

where

$$A = \frac{\prod_i^{3N} \nu_i^{reactant}}{\prod_i^{3N-1} \nu_i^{\ddagger}} \tag{3.2}$$

being E_a the minimum energy required to activate the chemical process, so the difference between the energy of the saddle point and the local energy of the initial configuration (reactant), ν_i, the corresponding normal mode frequency (‡ is referring to the transition state), k_B the Boltzmann constant and T the temperature.

3.3 Minimum Energy Path

From a chemical point of view the Minimum Energy Path (MEP) is the path lying on the potential energy surface between the reactant and product configurations, offering least resistance to the atomic motion.

Mathematically the MEP is the energy path between reactant (R) and product (P) coordinates going through the intermediate saddle point on the energy profile. From the saddle points, the path is the union of steepest descent paths to the minima. Assuming the potential energy function, $V(\vec{R})$, with the two minima V(R) and V(P), the MEP connecting these two states, R and P, is a smooth curve φ satisfying

$$(\nabla V)^{\perp}(\varphi) = 0, \tag{3.3}$$

where $(\nabla V)^{\perp}$ is the component of ∇V normal to φ,

$$(\nabla V)^{\perp}(\varphi) = \nabla V - (\nabla V, \widehat{\tau})\,\widehat{\tau}. \tag{3.4}$$

Here $\widehat{\tau}$ denotes the unit tangent of the curve φ, and $(\centerdot,\centerdot)$ denotes the Euclidean inner product. In appropriate mathematical setting, one can prove that the MEP is the most probable path that the system will take under the over-damped dynamics to move between R and P, crossing the barriers in-between.

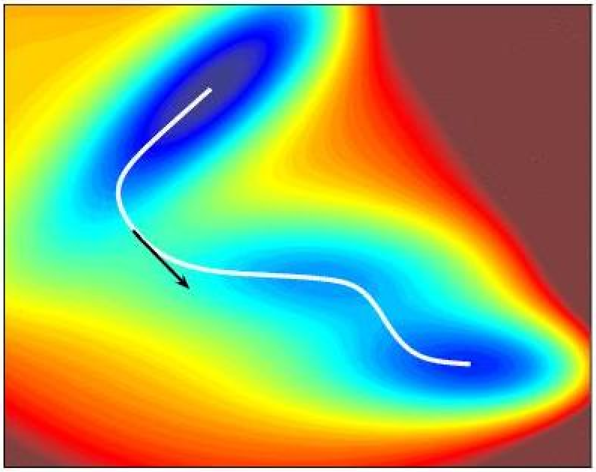


Figure 3.3: An example of a bi-dimensional smooth energy landscape and the minimum energy path (MEP) marked in white.

www.math.nus.edu.sg

It is important to underline that for the usual (bio)chemical systems, with a large number of atoms, hundreds or thousands, the potential energy surfaces are much more complex than the represented on the figure 3.3, due to the high dimensionality. This fact stands in the way of computing the minimum energy paths on the non trivial potential energy surfaces, often not known analytically. Therefore the need of introducing mathematical techniques to compute the geometrical MEP's becomes a fundamental task.

3.4 Methods for finding saddle points and MEPs

Many different methods have been developed for finding MEPs and saddle points. Such methods involve maximization of one degree of freedom, the reaction coordinate, and minimization in other degrees of freedom, since the saddle points to find are of first-order[1, 19].

The Drag method

It consists in holding one degree of freedom fixed, the so-called drag coordinate, and relaxing the other d-1 degrees of freedom, being d the dimension. The drag coordinate is increased stepwise and fixed while the other degrees of freedom are relaxed. The choice of the drag coordinate can be an intuitive guess or the straight line interpolation between the reactant and the product figure 3.4). Starting from the initial state the inverted force acting on the system along the drag coordinate is chosen and the velocity Verlet algorithm with a projected velocity simulates the dynamics of the system, till the saddle point is reached.

One important disadvantage of this method is the choice of the drag coordinate. This choice is crucial, because there are many cases in which the method fails, confining the solution in an energy valley, due to the drag coordinate is at a large angle to the direction of the saddle point.

The NEB method

In the Nudged Elastic Band (NEB) method, which is a chain method[20], a string of replicas (images) of the system between the reactant and product is created and the images are connected with springs; an optimization algorithm is applied to relax the images down

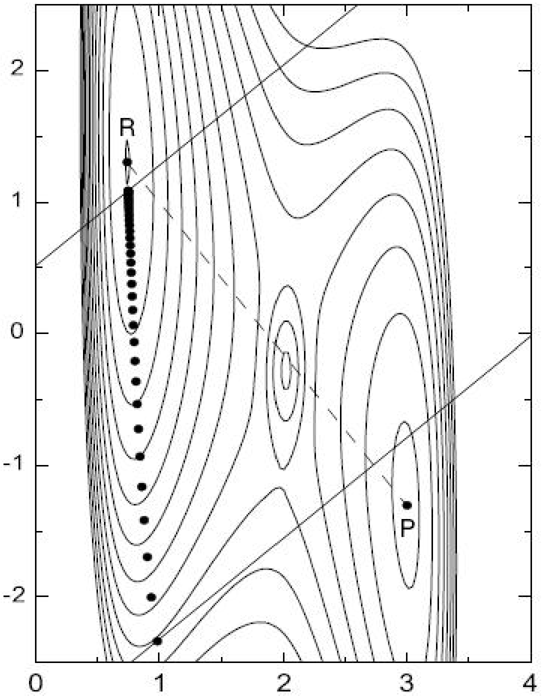


Figure 3.4: Schematic description of the Drag algorithm appears. It consists on fixing one degree of freedom which is increased stepwise while relaxing the other degrees of freedom. In this case the method cannot locate the saddle point.

Methods for finding saddle points and minimum energy paths;
G.Henkelman, G. Jóhannesson and H. Jónsson

towards the MEP, using the parallel spring force and the perpendicular component of the true force of every image, respect a vector defined depending on the path connecting the images and the energy values at every of these images (figure 3.5). At the end the replicas are located on the MEP, allowing to interpolate between such points and therefore obtaining the saddle point.

It is not going to give only the estimation of the saddle point but a global overview of the energy landscape.

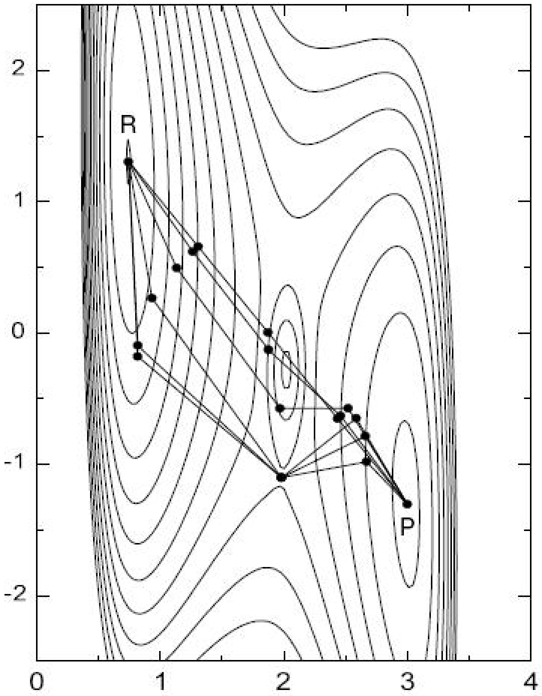


Figure 3.5: Schematic description of the NEB algorithm, where the guessed initial pathway, the line between R and P, and some other pathways obtained by different iterations are shown.

Methods for finding saddle points and minimum energy paths;
G.Henkelman, G. Jóhannesson and H. Jónsson

The CI-NEB method

It is a modified version of the NEB, named Climbing Image-NEB. The improvement consists of choosing, after some steps, the highest energy image and applying on it different forces than on the other images, only the inverted parallel component of the true force. Thanks to this modification, at the end of the method one of the images is exactly set in the saddle point.

The NEB method has been implemented and tested by us; a detailed description of our work will be presented in chapters 4 and 5.

The CPR method

It is the Conjugate Peak Refinement (CPR) method. As the NEB the CPR method also gives the MEP as final result. A set of images is generated, one at a time, between the initial and final configurations. The iterations involve the following steps:

- Set the vector between product, x_0 and reactant, x_{end}
- Find out the maximum on the direction of the vector $\Rightarrow y_i$
- Perform a minimization along all the conjugate vectors $\Rightarrow x_i$
- Repeat the procedure for $x_0 - x_i$ and $x_i - x_{end}$

It stops when the maximum along the path is characterized by a smaller gradient than the given tolerance need for finding a saddle point.

The Ridge method

The Ridge method sets two images and keeps moving in cycles of 'side steps' and 'downhill steps'. Firstly a straight line interpolation between the product, P, and the reactant, R, is performed and the maximum of energy along this line is found. Considering the maximum set at point a (figure 3.7). Then, two replicas of the system are created on the line, one on each side of the maximum, x'_0 and x'_1, the 'side steps', as it is shown in the figure 3.7. Now the force is evaluated at the two images and they are moved in the direction of their forces a certain distance, the 'downhill-step'. This generates points x''_0 and x''_1. First cycle is complete. A new cycle starts by maximizing along the line $[x''_0, x''_1]$ to obtain the point b, and so on. The side-step and downhill-step distance varies considering the proximity to a saddle point and the surrounding energy values.

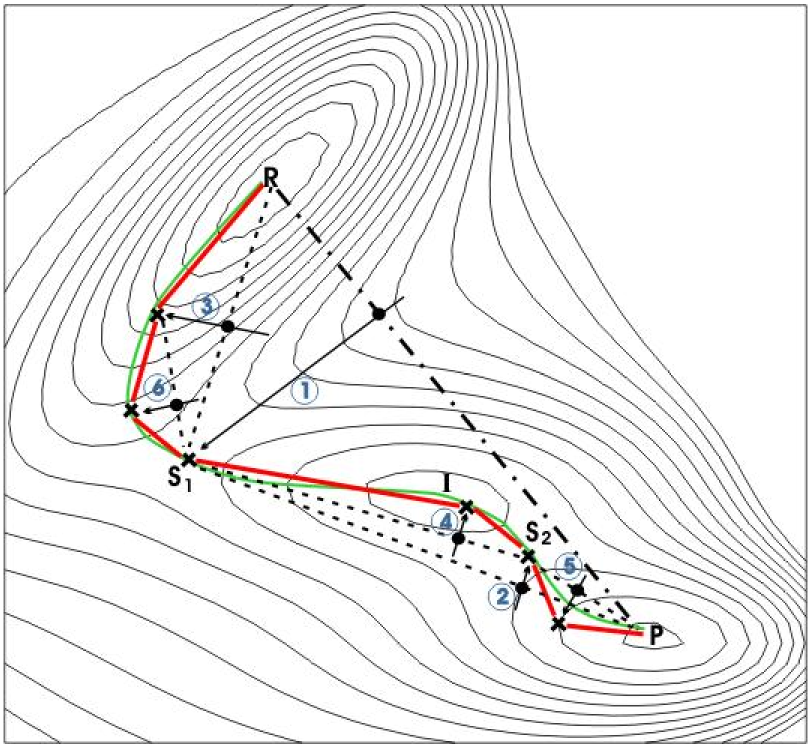


Figure 3.6: Schematic description of the Conjugate Peak Refinement (CPR) algorithm. The contour lines show the energy landscape. An initial guess of the path is made, here the straight interpolation line (-.-.-) from the reactant (R) to the product (P) states; at every iteration the maximum energy point on the line is marked by a dot and performed a conjugate minimization, obtaining a new configuration for the next line definitions.

http://spider.iwr.uni-heidelberg.de

The DHS method

Dewar, Healy and Stewart (DHS) have proposed a different method to compute the saddle point. First, the reactant R and product P are joined by a line segment. Every cycle has two main steps. In the first one the energy of both images is computed, and then the lower energy image is pulled towards the other image, through the line segment previously defined, in a percentage of 5%, obtaining a new image. Second, considering the new image and the initial one at higher energy, the energy of the actual lower energy image is minimized keeping the distance between the two fixed. This is repeated several times, reaching at the end the saddle point.

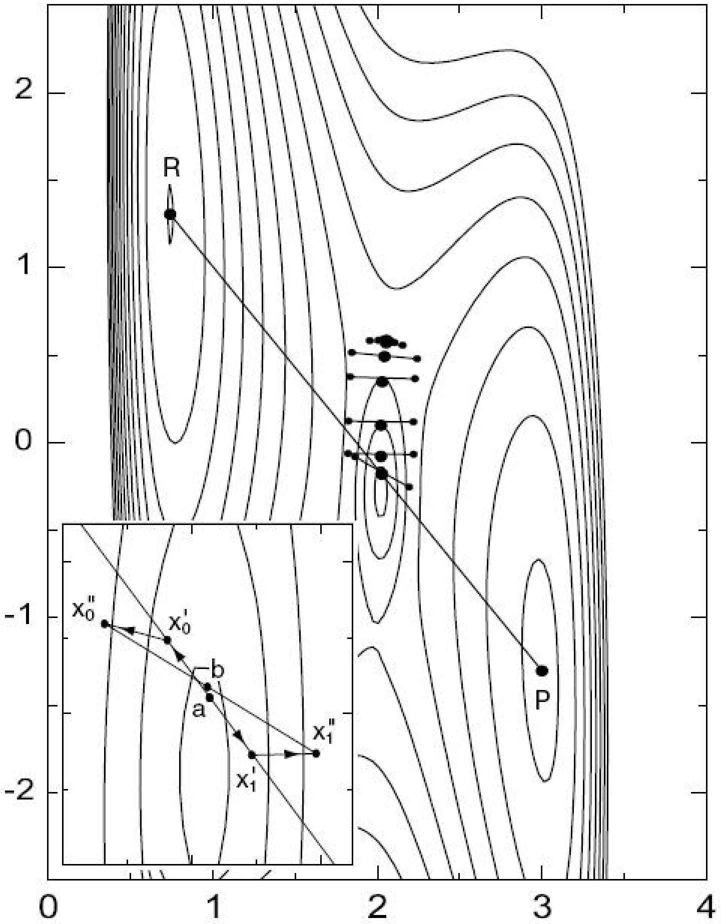


Figure 3.7: Schematic description of the Ridge algorithm. Starting with two images on each side of the potential energy ridge, they are moved towards the saddle point; firstly finding the maximum along the line from R to P, then defining two images, one on each side, x'_0, x'_1 which are displaced downhill along the gradient, obtaining x''_0, x''_1 and repeating this process until convergence to the saddle point.

Methods for finding saddle points and minimum energy paths;
G.Henkelman, G. Jóhannesson and H. Jónsson

The method can locate the neighbouring region of the saddle point quite quickly, but does not converge close to the saddle point efficiently.

It is necessary to use an optimization algorithm for the minimization of the position of the lower energy image.

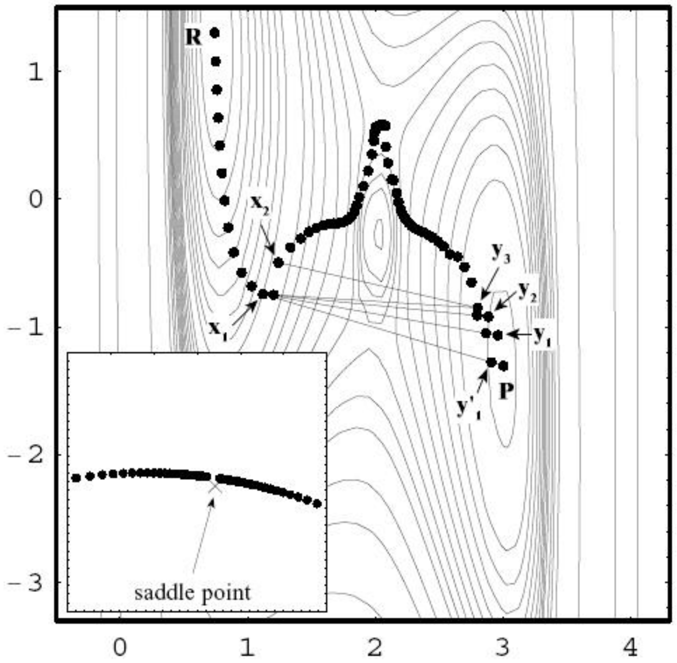


Figure 3.8: Schematic description of the Dewar, Healy and Stewart (DHS) algorithm. Starting with two images, R and P, a cycle consists on choosing the lower energy image and pulling towards the higher energy one, for after relax the lower energy but keeping the distance between the two new images. Eventually, the two images get close to the saddle point.

Methods for finding saddle points and minimum energy paths;
G.Henkelman, G. Jóhannesson and H. Jónsson

String method

The idea behind the string method is that a continuous reaction pathway is optimized to the MEP. The string method is very similar to the NEB. The initial pathway is defined by a set of images connected by linear segments. The same tangent vectors and perpendicular component of the true forces are computed, but there are not springs defined between every pair of images, so to achieve images they are distributed along the path at every step.

Chapter 4

NEB algorithm: our implementation

4.1 Introduction

The calculation of the activation Energy (e.g., the barrier height in a MEP, see figure 3.2) is a fundamental task when facing problems related with the kinetics of chemical reactions or diffusion events. As it was exposed in the previous chapter, there are several techniques to tackle the problem and perform the calculation, and in this master thesis the (CI-)NEB algorithm is the technique coded and used for computing the MEP on a model potential; the code will be used in the near future for quantum calculations.

Starting from Classical Mechanics, for computing the energy rate of transition or just the MEPs, it could be simulated the chemical or diffusion process attending to the dynamics described by the Newton's second law; but due to the actual computer capacity this is impossible, because in some concrete case this requires the order of 10^5 years in the fastest present day computer.

So the way to deal with this material problem is to obtain accurate estimates of transition rates using the results obtained by the algorithms to compute saddle points or MEPs and the Transition State Theory (TST).

The Nudged Elastic Band (NEB) and its modified version, the Climbing Image-NEB

(CI-NEB) will be explain on detail.

4.2 The Nudged Elastic Band method

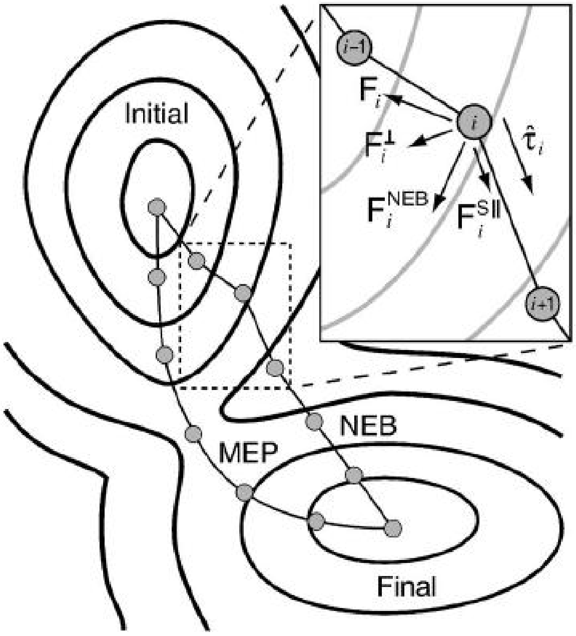


Figure 4.1: Here it is posed a general scheme of the NEB algorithm, where we can observe the initial guessed band, NEB, as a linear interpolation between the minimum points Initial and Final, the NEB force applied on each of the determined points in the band, called images and denoted by *i-1*, *i*, *i*+1, and the Minimum Energy Path (MEP), which is the goal of the NEB method.

The Journal of Chemical Physics **128**, 134106(2008)

Considering a chemical system, the initial state is the reactant configuration, $\vec{R}$, and the final configuration or the product, $\vec{P}$. They are known data, as the potential energy function V should be. For many cases in Quantum Mechanics is not possible to calculate the PES because of the very large number of degrees of freedom. The method defines a certain number N (typically $4-20$) of different molecular configurations (named images or replicas) in between and connects them with springs, creating a path from the reactant to the product configuration. Since we have only a limited information on the PES for concrete cases, the initial guess of such set of replicas can correspond to a linear interpolation between $\vec{R}$ and $\vec{P}$. Then an optimization algorithm is called for every image in order to compute the force acting on each of the images.

The image positions can be denoted by $\left[\vec{R}_0, \vec{R}_1, \vec{R}_2, ..., \vec{R}_N\right]$, where $\vec{R}_0 = \vec{R}$ and $\vec{R}_N = \vec{P}$.

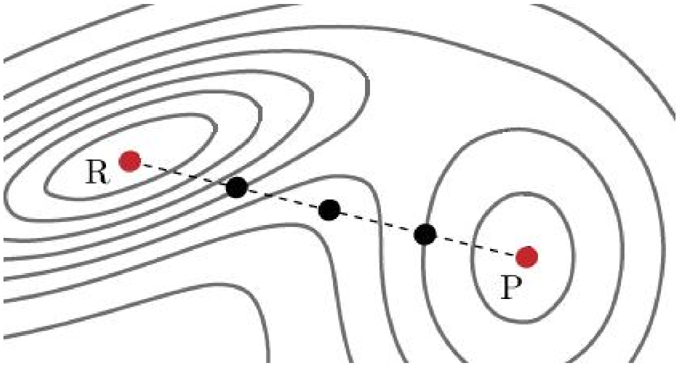


Figure 4.2: Sketch of the initial band, constructed connecting the initial state, $\vec{R}$, the final state, $\vec{P}$, and the images in between by a linear interpolation.

http://www.quantumwise.com

The original objective function proposed for the Elastic Band method [21] was

$$S(\vec{R}_1, \vec{R}_2, ..., \vec{R}_N) = \sum_{i=1}^{N-1} V(\vec{R}_i) + \sum_{i=1}^{N} \frac{k}{2}(\vec{R}_i - \vec{R}_{i-1})^2 \tag{4.1}$$

which was minimized respect to $\vec{R}_1, \vec{R}_2, ..., \vec{R}_N$, but this entailed two problems (figure 4.3) for the description of the MEP:

- The images tend to slide downhill, away from the saddle point representing the transition state. This can be reduced by using an appropriate stiffness for the spring constants.
- The band tends to cut corners in regions where the PES is curved. [19]

To face these two problems the objective function to minimize has been developed until becoming into the actual known NEB force.

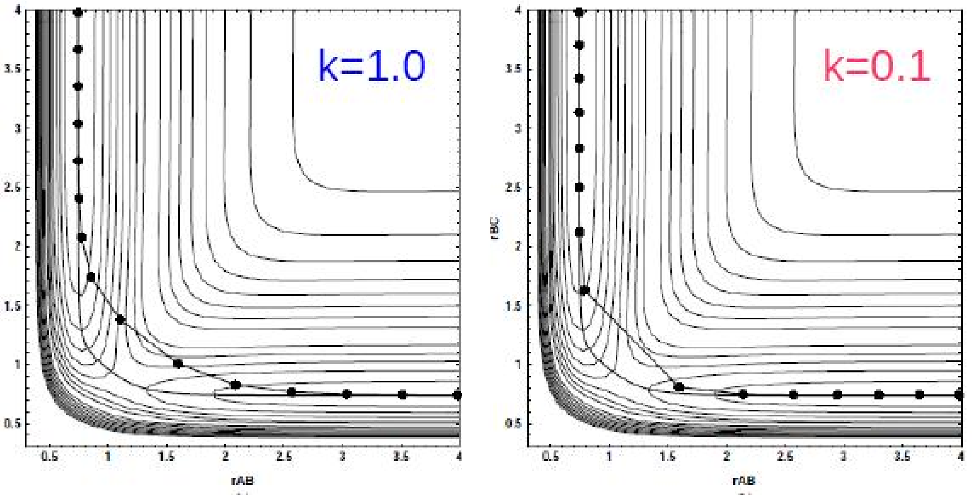


Figure 4.3: Potential Energy Surface (PES) where the 'corner-cutting' effect is observed on the left (the images do not pass through the MEP in the region with high curvature) consi, and the 'down-sliding' problem on the right, where the images tend to slide downhill, due to the objective function 4.1 to minimize proposed for the original Elastic Band method. The k represents the spring constant value.

The intermediate images $\vec{R}_1, \vec{R}_2, ..., \vec{R}_{N-1}$ are adjusted by a NEB force acting on each image i, which consists of the parallel spring force and the perpendicular component of the true force (e.g., the minus gradient of the potential V)[22]

$$\vec{F}_i^{NEB} = \vec{F}_i^{\perp} + \vec{F}_i^{S\parallel} \tag{4.2}$$

as it is shown in figure 4.1

In the NEB approach, special attention must be paid on avoiding all these images down-sliding to the initial and to the final state, for this reason the method applies to each replica a force perpendicular to the true force, being the true force the one following the energy gradient:

$$\vec{F}_i^{\perp} = -\nabla V(\vec{R}_i)|_{\perp} = -\nabla V(\vec{R}_i) + \nabla V(\vec{R}_i) \cdot \widehat{\vec{\tau}}_i \widehat{\vec{\tau}}_i \tag{4.3}$$

While for the corner-cutting problems the solution is to consider a complex parallel spring force, which also ensures equal spacing of the images. This spring force will be defined as

$$\vec{F}_i^{S\|} = k\left(|\vec{R}_{i+1} - \vec{R}_i| - |\vec{R}_i - \vec{R}_{i-1}|\right)\widehat{\vec{\tau}}_i \tag{4.4}$$

where k is a parameter given by input.

In order to have a better description of the saddle point, a larger density of images close to it could be desirable, such goal is achieved by a spring parameter k depending on the energy values [18] of the images:

$$k_i = \begin{cases} k_{max} - \Delta k \left(\dfrac{V_{max} - V_i}{V_{max} - V_{ref}}\right) & if \quad V_i > V_{ref} \\ k_{max} - \Delta k & if \quad V_i < V_{ref} \end{cases} \tag{4.5}$$

being

$V_i = max\,\{V_i, V_{i-1}\};$
$V_{max} = max\,\{V_i\};$
V_{ref}, k_{max} and Δk are input parameters depending on the specific nature of the PES under study.
Using the definition in equation 4.5, images with low energy are connected by a weaker spring constant, whereas images with higher energy are connected by stronger spring forces, being therefore closer each other and giving a more detailed description of the saddle point region.

In the Climbing Image NEB (CI-NEB) method, after a few iterations, the spring force is not applied to the highest energy image l, and it climbs to the saddle point via a reflection of the true force (determined by the gradient of the potential energy V) along the tangent $\widehat{\tau}_i$ [22]

$$\vec{F}_l^{CI} = \vec{F}_l - 2\vec{F}_l \cdot \widehat{\vec{\tau}}_i\widehat{\vec{\tau}}_i \tag{4.6}$$

For the successful computation of all these forces, defining the vector $\widehat{\vec{\tau}_i}$ is essential. $\widehat{\vec{\tau}_i}$ is the tangent vector to the path, referred to the image i.

4.3 Estimate of the tangent

The determination of the tangent to the path is crucial to get reliable an accurate results. The original implementation of the NEB method considered as tangent to the i-th-replica $\vec{R}_i$ the following normalized line segment, estimated taking into account the adjacent images $\vec{R}_{i-1}$ and $\vec{R}_{i+1}$

$$\widehat{\vec{\tau}_i} = \frac{\vec{R}_{i+1} - \vec{R}_{i-1}}{|\vec{R}_{i+1} - \vec{R}_{i-1}|} \tag{4.7}$$

A slight improvement to ensure equal spacing between images even in regions with large curvature, by using a simple bisection formula, is given by [19]:

$$\vec{\tau}_i = \frac{\vec{R}_i - \vec{R}_{i-1}}{|\vec{R}_i - \vec{R}_{i-1}|} + \frac{\vec{R}_{i+1} - \vec{R}_i}{|\vec{R}_{i+1} - \vec{R}_i|} \tag{4.8}$$

The vector $\vec{\tau}_i$ is then normalized, $\widehat{\vec{\tau}_i} = \dfrac{\vec{\tau}_i}{|\vec{\tau}_i|}$

Anyway, these two estimates are not working in some cases, as when the energy of the system changes rapidly along the path or covalent bonds are broken and formed in the case of chemical reactions. In these problematic cases there are angles formed by $\vec{R}_i - \vec{R}_{i-1}$ and $\vec{R}_{i+1} - \vec{R}_i$ far from zero value, obtaining a parallel component of the force larger than the perpendicular one and this creates kinks on the band (figure 4.4) for the next iterations not allowing convergence to the MEP. To avoid such drawbacks a new tangent is defined, involving the energy values of the image i and of the adjacent replicas [23]:

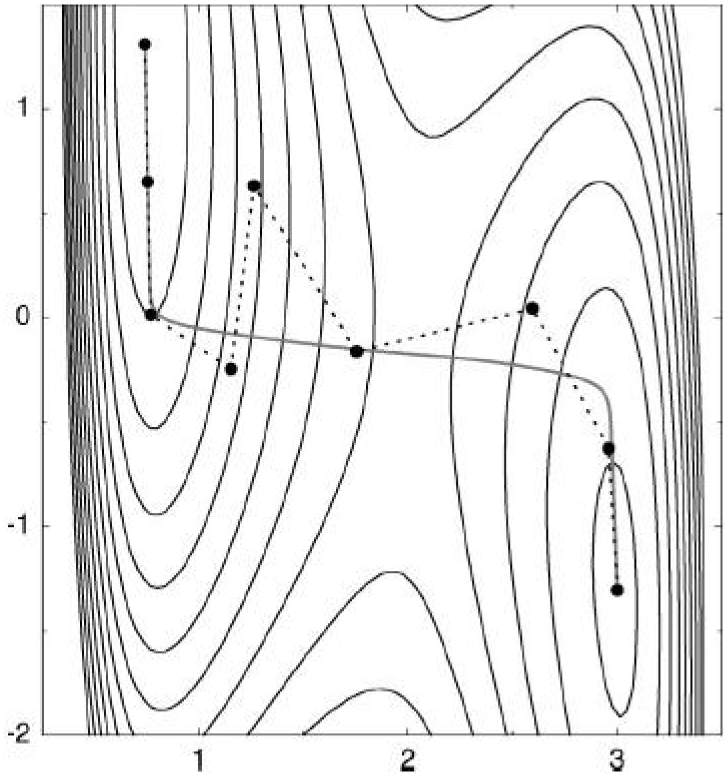


Figure 4.4: The original nudged elastic band method as described by equations 4.7 and 4.8 can generate kinks along the path as illustrated here by the dashed line and perpetuate in time, not converging to the Minimum Energy Path (continuous line).

Journal of Chemical Physics **113**, 9978 (2000)

$$\vec{\tau}_i = \begin{cases} \vec{\tau}_i^{+} & if \quad V_{i+1} > V_i > V_{i-1} \\ \vec{\tau}_i^{-} & if \quad V_{i+1} < V_i < V_{i-1} \end{cases} \tag{4.9}$$

with

$$\vec{\tau}_i^{+} = \vec{R}_{i+1} - \vec{R}_i$$
$$\vec{\tau}_i^{-} = \vec{R}_i - \vec{R}_{i-1}$$

and if the image is at a minimum or at a maximum in energy

$$\vec{\tau}_i = \begin{cases} \vec{\tau}_i^{+} \Delta V_i^{max} + \vec{\tau}_i^{-} \Delta V_i^{min} & if \quad V_{i+1} > V_{i-1} \\ \vec{\tau}_i^{+} \Delta V_i^{min} + \vec{\tau}_i^{-} \Delta V_i^{max} & if \quad V_{i+1} < V_{i-1} \end{cases} \tag{4.10}$$

where

$$\begin{aligned} \Delta V_i^{max} &= max\,(\mid V_{i+1} - V_i \mid, \mid V_{i-1} - V_i \mid) \\ \Delta V_i^{min} &= min\,(\mid V_{i+1} - V_i \mid, \mid V_{i-1} - V_i \mid) \end{aligned}$$

and the final tangent is the normalized one

$$\widehat{\vec{\tau}}_i = \frac{\vec{\tau}_i}{|\vec{\tau}_i|}.$$

4.4 Optimization of the force

Once the tangent at every image is determined, it is possible to define the parallel spring force and the perpendicular component of the true force according to equation 4.2, the NEB force responsible for the convergence to the MEP. For the spring force all the elements are fully determined, but for computing the true force and applying the final NEB force a minimization method must be used, the same for all the replicas.

The minimization algorithm evaluates the energy and the gradient (true force) for each image having as input the position of the image i in the configuration space of the system. The NEB algorithm then, for each image, using the coordinates and energy of the two adjacent images, estimates the local tangent to the path, and the spring constant in the required case. Thanks to these data the perpendicular component of the true force respect the tangent and the parallel spring force can be computed, giving the NEB force. The NEB force is applied to every image in the minimization algorithm, setting the new positions for the following iteration in the NEB algorithm.

The process continues until absolute value of the maximum component of the NEB force at each image is less than a fixed initial tolerance.

$$|\vec{F}^{NEB}_{max}| < \varepsilon \tag{4.11}$$

When this tolerance is achieved all the optimization algorithms stop and the final result corresponds to the converged MEP (figure 4.5).

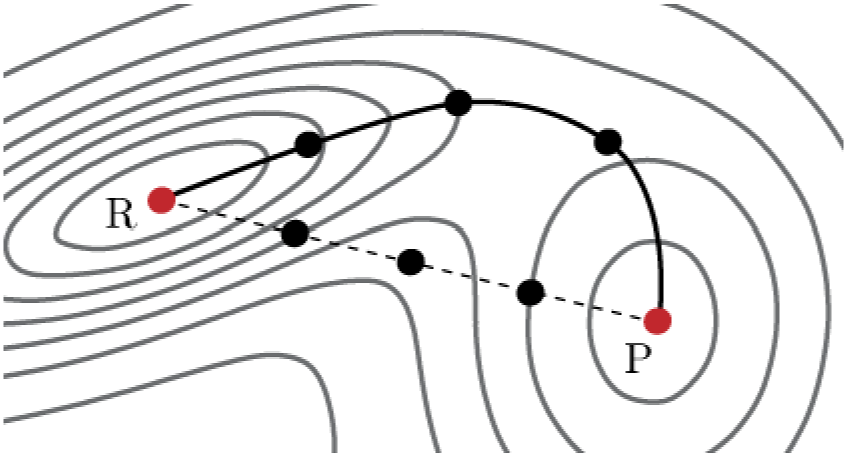


Figure 4.5: Sketch with the initial and final states, R and P, connected by the initial band as a dashed line and by the obtained MEP as a solid line.

http://www.quantumwise.com

4.5 Interpolation

The final step is to locate the saddle point and to plot the MEP profile, and for achieving that it is necessary to interpolate between the final images of the converged elastic band. The energy of the images and the force along the band provides the extra information that should be used to interpolate. A cubic polynomial, $ax^3 + bx^2 + cx + d$, is used for the interpolation to each segment $[\vec{R}_i, \vec{R}_{i+1}]$, so setting each image in the segment as an unidimensional point, $\vec{R}_i \equiv 0$ in the coordinates origin, and $\vec{R}_{i+1} \equiv R_* = |\vec{R}_{i+1} - \vec{R}_i|$, we obtain the system

$$\begin{cases} d = V_i \\ aR_*^3 + bR_*^2 + cR_* + d = V_{i+1} \\ c = -F_i \\ 3aR_*^2 + 2bR_* + c = -F_{i+1} \end{cases} \tag{4.12}$$

with the parameters given by

$$\begin{aligned} a &= \frac{2\left(V_i - V_{i+1}\right)}{R_*^3} + \frac{F_i + F_{i+1}}{R_*^2} \\ b &= \frac{2F_i - F_{i+1}}{3R_*} + \frac{3\left(V_{i+1} - V_i\right)}{R_*^2} \\ c &= -F_i \\ d &= V_i \end{aligned} \tag{4.13}$$

where V_i, V_{i+1} are the energy values and F_i, F_{i+1} are the value of the forces along the path for images i and $i+1$ respectively.

It could be possible also to generate a quintic polynomial interpolation forcing the second derivative to be continuous.

4.6 Implementation

After the detailed explanation about the Nudged Elastic Band method it is the time to use it. We have developed a complex codification using some different scientific programming codes, as Fortran, C and Python.

Algorithm

First of all we state, in a schematic way, the steps for the algorithm:

1. Set the desired initial and final atomic configurations, $\vec{R}$ and $\vec{P}$.

2. Construct an initial reaction pathway between $\vec{R}$ and $\vec{P}$. The reaction configurations can be denoted by $\left[\vec{R}_0, \vec{R}_1, \vec{R}_2, ..., \vec{R}_N\right]$, where $\vec{R}_0 = \vec{R}$ and $\vec{R}_N = \vec{P}$. They are chosen by a linear interpolation between $\vec{R}$ and $\vec{P}$ or according to the user's strategy.

3. Compute the energy $V(\vec{R}_i)$, and the force acting on every image defined by the gradient of the potential energy:
$\vec{F}_i = -\nabla V(\vec{R}_i)$.

4. Compute the tangent $\widehat{\vec{\tau}}_i$ to the pathway at each image.

5. Connect each pair of images with a spring, yielding a force on every image of
$$\vec{F}_i^{S\|} = k\left(|\vec{R}_{i+1} - \vec{R}_i| - |\vec{R}_i - \vec{R}_{i-1}|\right)\widehat{\tau}_i \ .$$

6. Project out the component of the true force parallel to the tangent at every image i,
$\vec{F}_i^{\perp} = -\nabla V(\vec{R}_i) + \nabla V(\vec{R}_i) \cdot \widehat{\vec{\tau}}_i \widehat{\vec{\tau}}_i$.

7. Minimize the energy for every image using the NEB force, $\vec{F}_i^{NEB} = \vec{F}_i^{\perp} + \vec{F}_i^{S\|}$, as explained before, obtaining the new relaxed images. This step, as the third one, is performed by an energy minimizer algorithm.

   Then, steps from 3 to 7 will be repeated until getting a NEB force smaller than a tolerance, according to the condition in equation 4.11.

8. Cubic polynomial piecewise interpolation of the final images.

Implementation characteristics

The main features of the implementation are:

- It is done in such a general way as to allow us using it with whatever optimization code we want to run in steps 3 and 7. Just by including a few generalized lines of code, and commenting the stop criterion on the minimization programs, you can use any Fortran code minimizing the energy (quantum codes for instance) in our general machinery and perform the NEB convergence independently of the minimizer employed and the level of theory used. We have used a home-made code implementing the Steepest Descent algorithm.

- The skeleton of the NEB algorithm and the full steepest descent algorithm have been implemented in Fortran language, while the part corresponding to communication between the main algorithm (the NEB one) and the optimization code, running for every image, are provided by some C libraries, named libmsock, interfaced with the Fortran source. The necessary files to run the algorithms have been created using Python codes and bash scripts are eventually used for creating folders, like the coordinate files for the image,) and running the programs.

- Everything is implemented based on a client/server model (figure 4.6), where the $N-2$ clients correspond to the minimizers optimizing the position of all the images (one-to-one correspondence between image and client process) and the server role is played by a Fortran code which calculates the NEB force (equation 4.2). It is a model one-to-all.

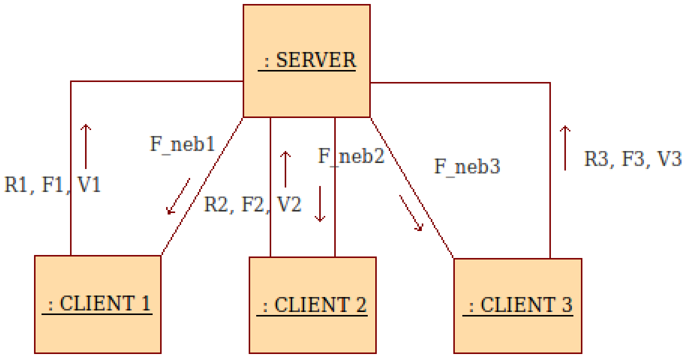


Figure 4.6: Scheme of the server-client model, where the server performs the general steps of the NEB algorithm while each client runs a minimization code for an image.

Lines added to the minimization code, where we manage the sockets for communicating each client and send the true force and energy value to the server in order to receive from this one the NEB force to set the new position:

```
!***************************************************************
!          NEB INSERTION - Opening socket                   Begin
!***************************************************************
sockfd=ClientSocket(hostname,port)
IF(sockfd<0) then
          write(*,*) 'Failed␣to␣open␣client␣socket␣with␣port␣',port
          STOP
else
          write(*,*) 'Open␣client␣socket␣with␣port␣', port
end if
!***************************************************************
!          NEB INSERTION - Opening socket                     End
!***************************************************************


!***************************************************************
!          NEB INSERTION - Sending and receiving information Begin
!***************************************************************
          !Sending position, force and energy to the neb algorithm
          write( buffer, * ) R1

          !write(*,*) 'Sending coordinates', R1, 'to server'
          error=sockPuts(sockfd,buffer//C_NEW_LINE)
          if(error<0) STOP "Server␣died!"
          write( buffer, * ) F1

          !write(*,*) 'Sending forces', F1, 'to server'
          error=sockPuts(sockfd,buffer//C_NEW_LINE)
          if(error<0) STOP "Server␣died!"
          write( buffer, * ) V1
          !write(*,*) 'Sending energy', V1, 'to server'
          error=sockPuts(sockfd, buffer//C_NEW_LINE)
          if(error<0) STOP "Server␣died!"

          !Receiving the NEB force into the force variable
          length=sockGets(sockfd, buffer, INT(LEN(buffer), C_INT))
          if(length>0) then
                    read(buffer,*) F1
                    write(*,*) 'Received␣NEB␣force', F1, 'from␣server'
          endif
!**************************************************************
!          NEB INSERTION - Sending and receiving information          End
!**************************************************************


!**************************************************************
!          NEB INSERTION - Closing socket                            Begin
!**************************************************************
error=CloseSocket(sockfd)
write(*,*)'Client␣communication␣closed'
write(*,*) ''

IF(error<0) STOP "Failed␣to␣close␣server␣socket"
!**************************************************************
!          NEB INSERTION - Closing socket                              End
!**************************************************************
```

- For allowing the interprocess communication between server and clients we create sockets. A socket provides a bidirectional communication endpoint for sending and receiving data with another socket. The socket communication can work on different hosts but our tests have been carried out on the same machine.

The header of the C library used for defining socket functions is the following:

```
/*
 *  socket utility routines
 *
 *  RCS:
 *      $Revision: 1.1.1.1 $
 *      $Date: 1997/05/26 22:43:03 $
 *
 *  Description:
 *      most of the functions are taken (in some cases modified) from the
 *      Internet Socket Programming FAQ example code.
 *
 *  Development History:
 *      who                  when         why
 *      muquit@semcor.com    21-Mar-96    first cut
 */

#ifndef SOCKHEAD_H
#define SOCKHEAD_H 1

/* socket functin prototypes
*/
int sockRead        (int,char *,size_t);
int sockWrite       (int,char *,size_t);
int sockWriteDouble        (int,double *,size_t);

int ClientSocket   (char *netaddress,u_short port);
int ServerSocket   (u_short,int);
int getConnection  (int, u_short,int *,int);
int makeConnection(char *,int,char *);

int sockGets       (int,char *,size_t);
int sockPuts       (int,char *);
int sockPutsDouble        (int,double *);
int getHostByName  (char *host_found,char *check_for);

int getPeerInfo    (int,char *,char *,u_short *);

int atoport        (char *,char *);
int iread          (int,char *,int);



struct in_addr *atoaddr(char *);

#endif /* SOCKHEAD_H*/
```

We already presented the sockets opening on each client, as the extra lines in the optimization code, and here there are the lines opening the server sockets and how they are used to send the NEB force to the clients and to receive information from the clients:

```
!Define, open and bind the server sockets
do i=1,m              !m is the number of images between R and P
          port=5000+i
          WRITE(*,*) "Connecting_to_port",port
          Tot_sockfd(i) = ServerSocket(port,2_c_int)
          WRITE(*,*) "Client_connected!", i
          IF(ToT_sockfd(i)<0) STOP "Failed_to_open_server_socket"
end do
!**********************************************************
!Getting coordinates from each client (image)
```

```
do i=1,m
        sockfd = Tot_sockfd(i)

        !Getting coordinates
        length=sockGets(sockfd,ToT_buffer(i),INT(LEN(ToT_buffer(i)),C_INT))

        IF(length<0) EXIT        !Here we will keep the last values
        IF(length>0) read(ToT_buffer(i),*) ToT_R(i+1,:,:)

        !Getting forces
        length=sockGets(sockfd,ToT_buffer(i),INT(LEN(ToT_buffer(i)),C_INT))

        IF(length<0) EXIT        !Here we will keep the last values
        IF(length>0) read(ToT_buffer(i),*) ToT_F(i,:,:)
        !write(*,*) 'F',i, Tot_F(i,:,:)

        !Getting energies
        length=sockGets(sockfd,ToT_buffer(i),INT(LEN(ToT_buffer(i)),C_INT))

        IF(length<0) EXIT        !Here we will keep the last values
        IF(length>0) read(ToT_buffer(i),*) ToT_E(i+1)
        !write(*,*) 'E',i+1, Tot_E(i+1)

enddo
!*************************************************************
!Sending the NEB force to the clients
do i=1,m
        write(ToT_buffer(i), * ) F_neb(i,:,:)
        !write(*,*) 'Sending', F_neb(i,:,:), 'to client',i
        error=sockPuts(Tot_sockfd(i),ToT_buffer(i)//C_NEW_LINE)
        IF(error<0) STOP "Server␣died!"
enddo
!*********************************************************
```

Flowchart and details

The server executes the main instructions. It is the software program creating a socket for every client, controlling the NEB method flow and once the convergence is reached closing all the sockets and stopping the program.

Every client corresponds to an image, and opens a different socket used already by the server, in order to create the connection with the server. Each client runs the energy optimizer algorithm (in the case of our proofs, the steepest descent) to get the force applied to the reaction coordinates, and the energy value. No exit control is applied to the clients, since the threshold for convergence has been chosen to be related to the NEB force: $|\vec{F}_{max}^{NEB}| < \varepsilon$

A crucial point is to implement the communication between server and clients. Each socket has a unique address that is the combination of an IP (Internet Protocol) address and a TCP/IP (Transmission Control Protocol/Internet Protocol) port number and it is going to be connected to the minimization code. Since all, server and clients are running in parallel as independent processes which exchange information with the server, the sockets are used to transfer the information passing from server to clients and vice versa.The server needs true force and energy from every client at every iteration to compute the NEB force, while the clients need the NEB force at every iteration of the steepest descent algorithm to set the new position.

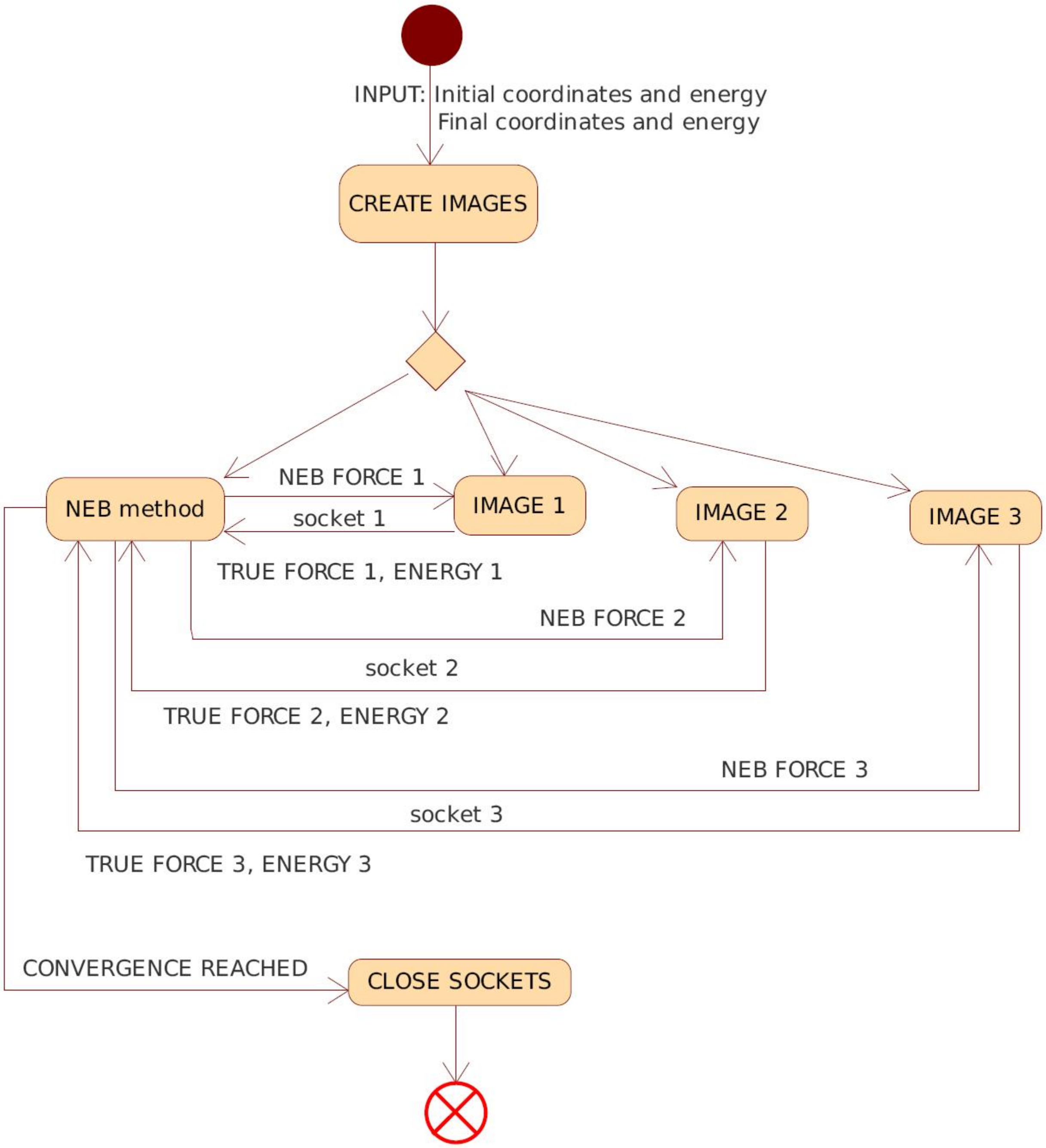


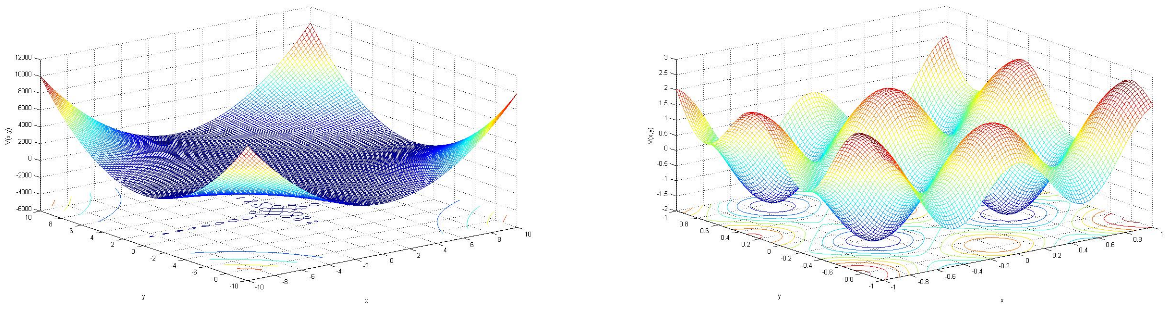
Figure 4.7: The general flowchart of the NEB implementation.

Chapter 5

Results

In this section we report the results obtained with the software we have developed, showing the robustness of the NEB algorithm, how it works in extreme cases, with bad initial guessing MEP or adding noise to the computed forces by the SD algorithm. Since we want to visualize the results, we are not going to use the energy equations explained in section 1.2 and section 1.3, instead we are going to use a model energy surface defined for only two coordinates.

$$V\left(x,y\right)=\cos\left(2\pi x\right)+\sin\left(2\pi y\right)+\left(xy\right)^{2} \tag{5.1}$$



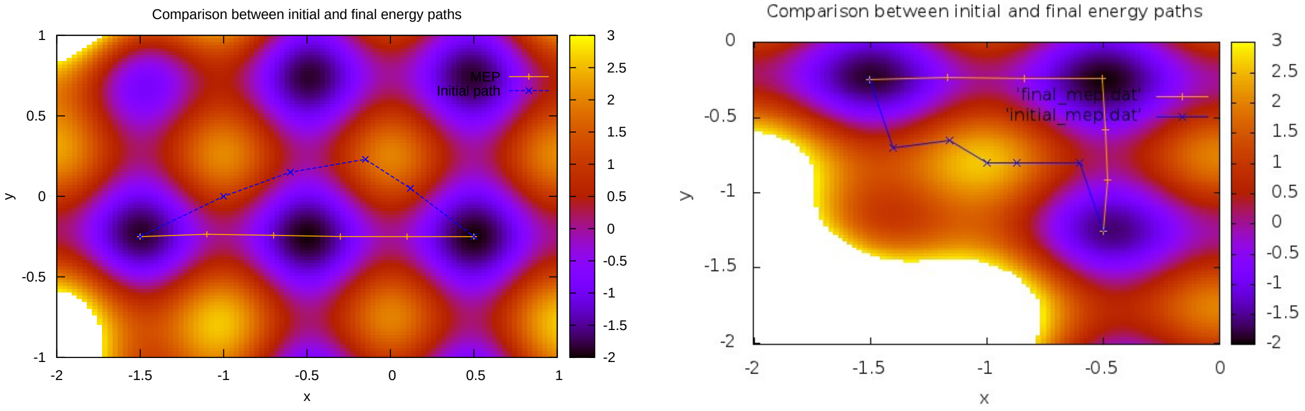
(a)

(b)

Figure 5.1: The used model potential energy surface, where (a) is a general overview of the energy surface, defined in the interval $[-10,10]\times[-10,10]$, and in the image (b) appears a detail of the energy function defined in $[-1,1]\times[-1,1]$

5.1 Results with no noise

We have performed several tests to debug and validate our approach, starting from a very simple case with two equivalent minima (reactant and product) and only one saddle point. Then we have moved to more interesting cases, in which the MEP is characterized by an intermediate minimum between the given initial and final positions.



(a)

(b)

Figure 5.2: Two different examples are shown, one in (a), where there is a minimum position located in the segment line fixed by the initial and the final positions, and other in (b), where the three minimum positions are forming a right angle. The initial guess of the MEP is not a linear interpolation but a more distorted curve.

Fixing from now on the parameter $h = 0.005$ for the Steepest Descent (equation 2.4), which defines the size of the displacement.

The data in table 5.1 refer to the case of the three minima in line, reporting the convergence properties (in terms of iterations) as a function of the NEB threshold ε (equation 4.11) and the spring constant; and the plot in figure 5.3 sketches how the iterations vary depending on these two parameters. We can observe, in general, a faster convergence when using bigger spring constants k.

Iterations				
NEB threshold	$k = 0.1$	$k = 1$	$k = 10$	$k = 100$
0.01	1095	740	195	45
0.005	1070	1070	231	48
0.001	7001	1905	314	56
0.0005	7001	2268	350	60

Table 5.1: Number of NEB iterations for the case of three minimum energy configurations in line (case (a) in figure 5.2), considering different values for the stiffness of the spring constants to define equation 4.4 and the NEB threshold ε appearing in equation 4.11.

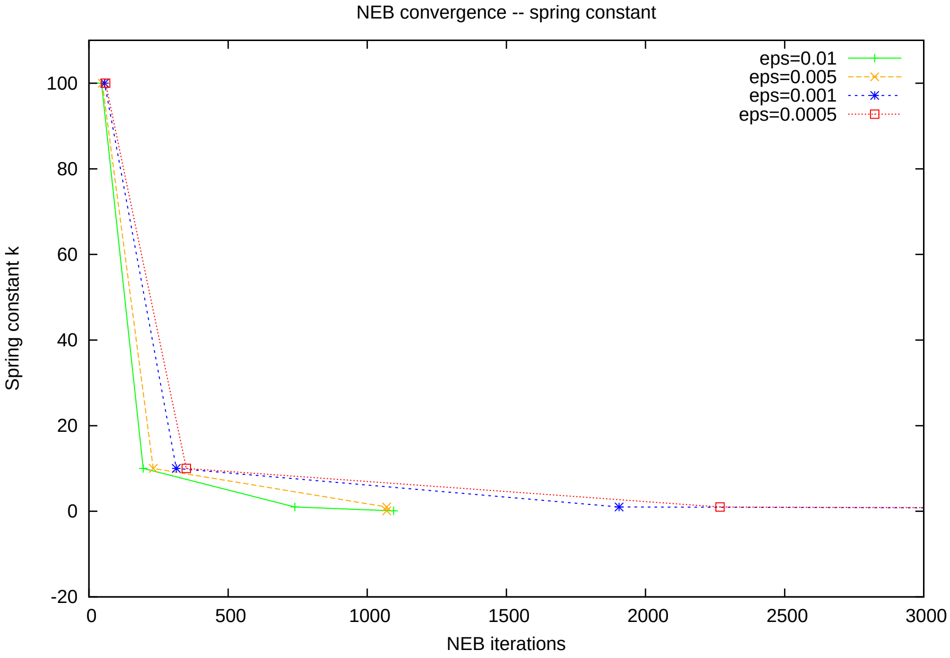


Figure 5.3: Plot of the iteration evolution as a function of the spring constant k for the case of 3 minima located in line, (a) in figure 5.2, given different values for ε.

The MEP resulting after running the NEB code is described in figures 5.4 and 5.5 for the case (a) in figure 5.2.

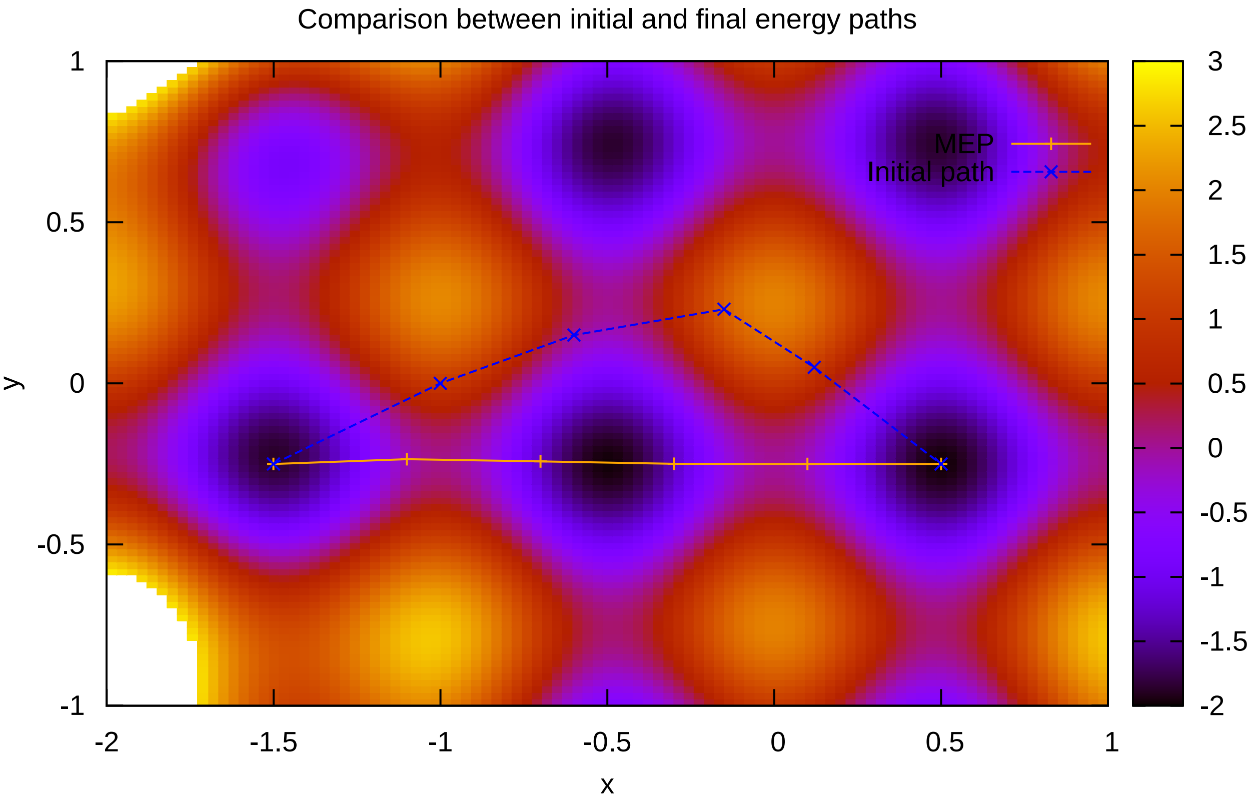


Figure 5.4: Projection of the initial pathway (dashed blue line) between the two minima, and the MEP (orange line) obtained after running the NEB algorithm for the case (a) in figure 5.2.

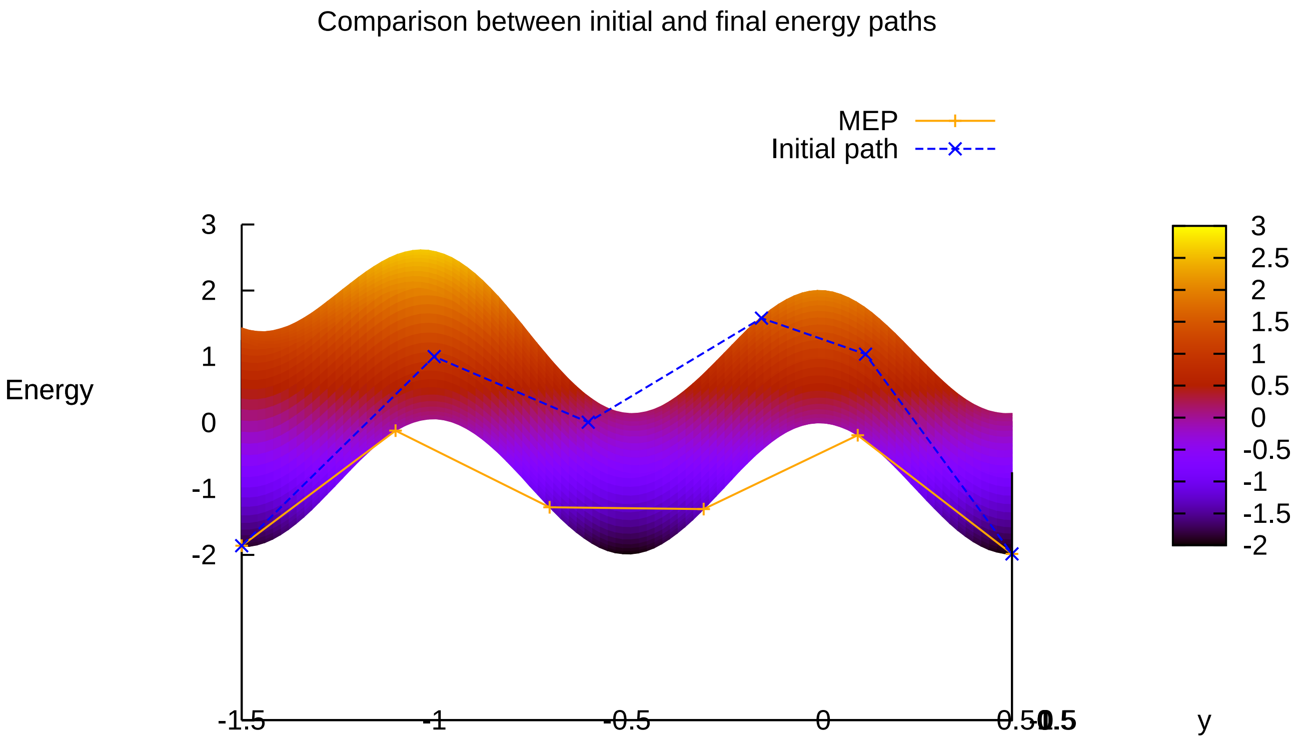


Figure 5.5: Profile of the initial pathway (dashed blue line) between the two minima, and the final MEP (orange line)for the case (a) in figure 5.2.

And analogously, the data for the case (b) in figure 5.2 are set in table 5.2 and in figure 5.6. While the initial pathways and the resulting MEPs obtained with the NEB algorithm are shown in figures 5.7 and 5.8.

Iterations				
NEB threshold	$k = 0.1$	$k = 1$	$k = 10$	$k = 100$
0.01	254	1141	302	62
0.005	1256	1642	358	66
0.001	7001	2893	493	74
0.0005	7001	3460	551	78

Table 5.2: Number of NEB iterations for the case of three minimum energy configurations forming a right angle (case (b) in figure 5.2), considering different values for the stiffness of the spring constants to define equation 4.4 and the NEB threshold ε appearing in equation 4.11.

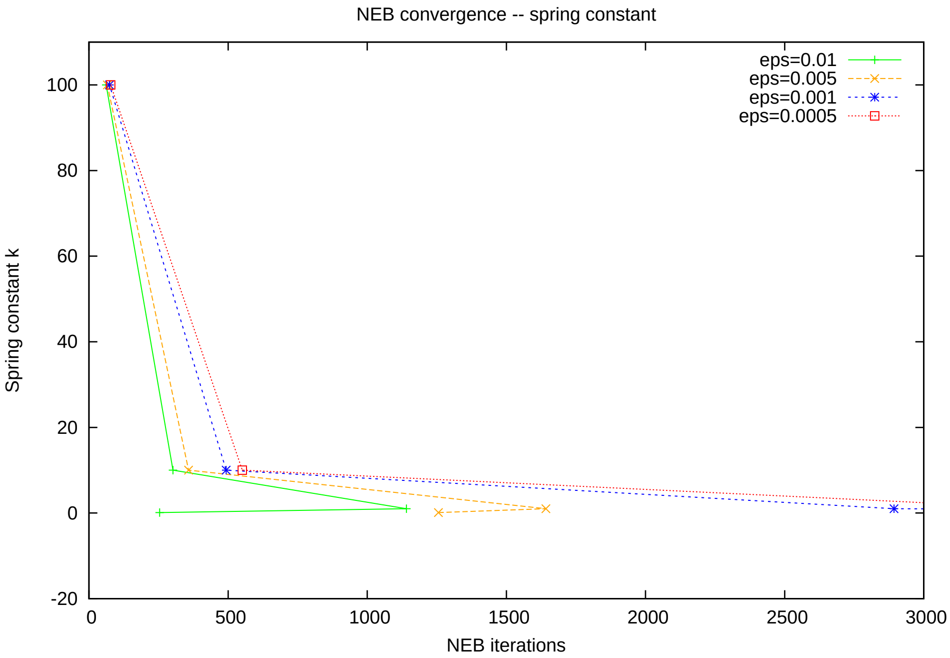


Figure 5.6: Plot of the iteration evolution as a function of the spring constant k for the case of 3 minima located in a right angle, (b) in figure 5.2, given different values for ε.

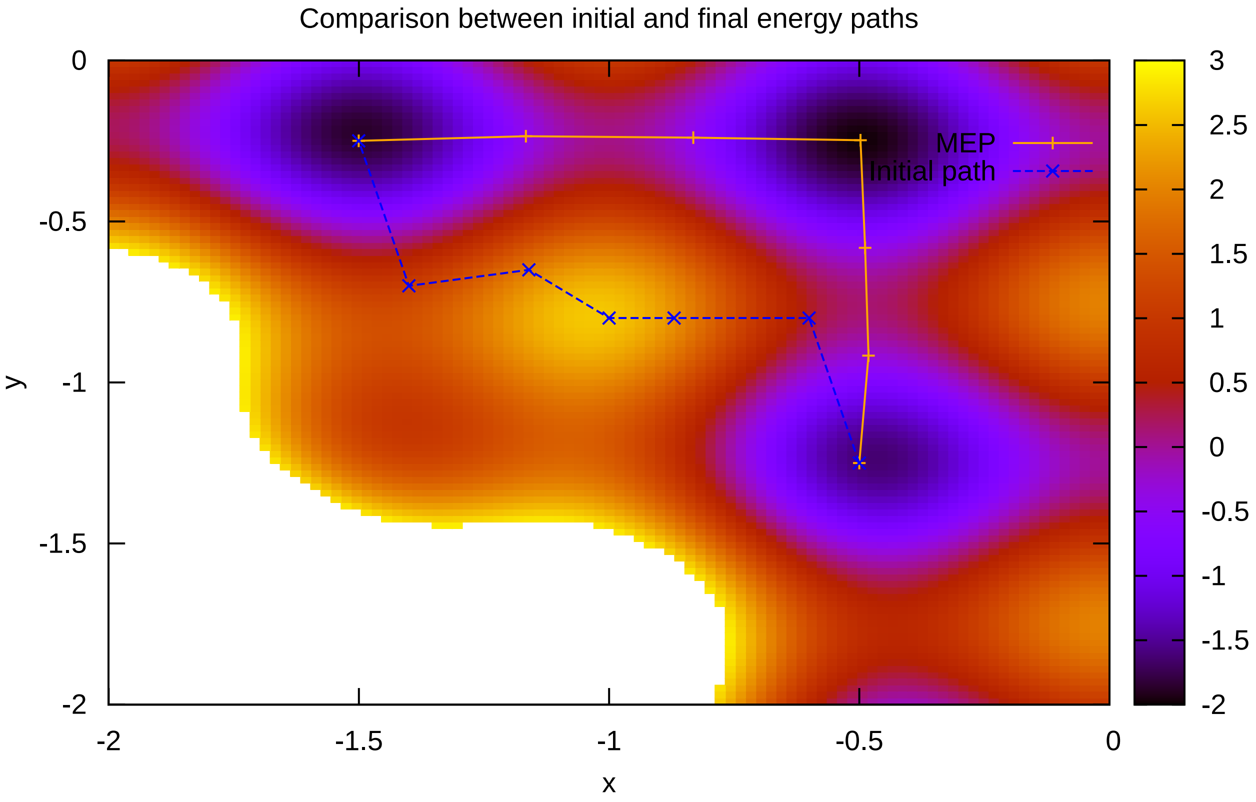


Figure 5.7: Projection of the initial pathway (dashed blue line) between the two minima, and the MEP (orange line) result from the NEB implementation in the case (b) of figure 5.2.

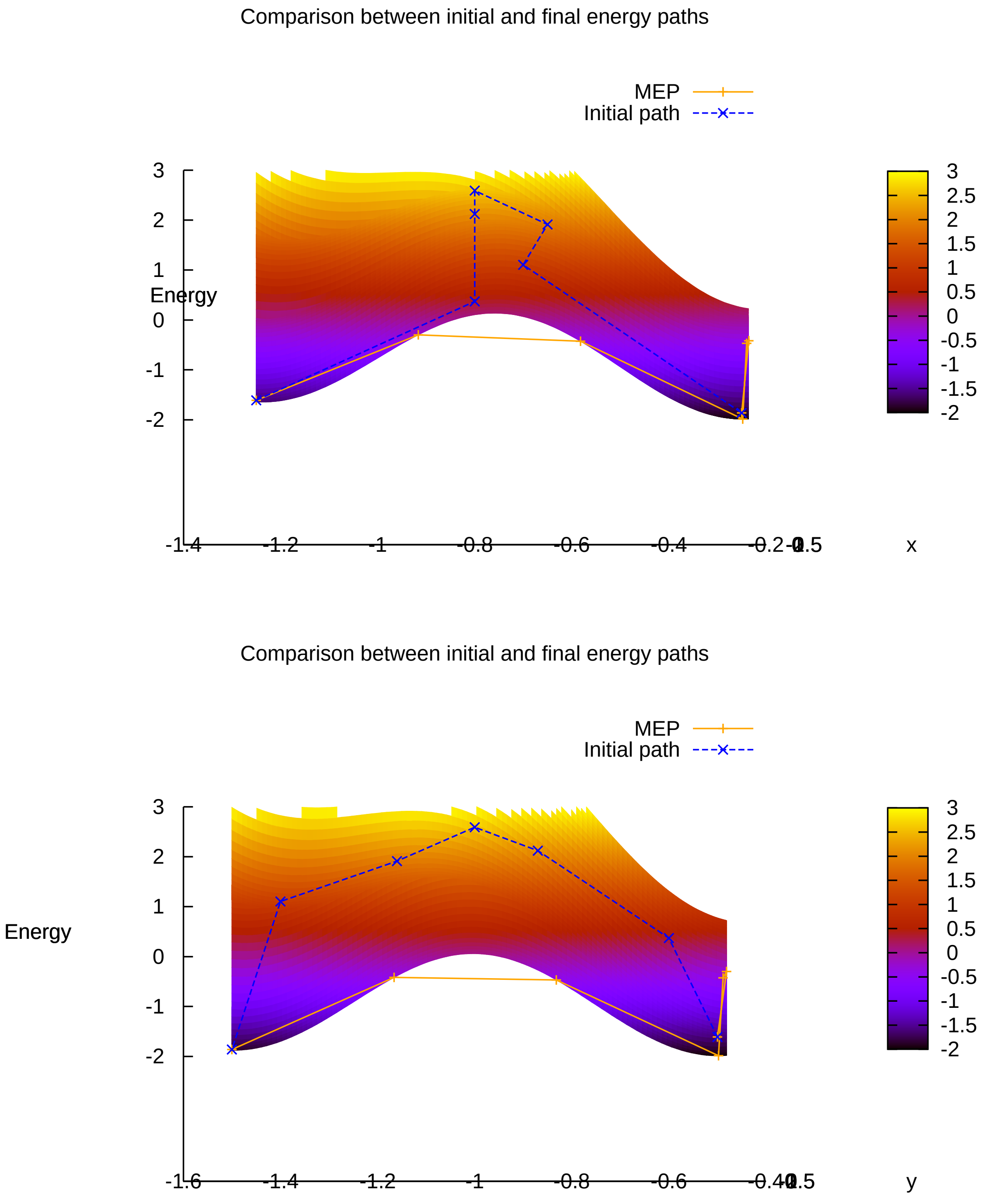


Figure 5.8: Profiles of the initial pathway (dashed blue line) between the two minima, and the MEP (orange line) for the example (b) in figure 5.2.

A clear convergence to the MEP is observed in both cases; the two saddle points and the intermediate minimum are easily recognized; but there is no point going to coincide with the saddle points, and the estimation of the potential barrier could be poorly accurate; one could increase the number of replicas (with the risk of kinks in the band) or use an energy-dependent k (equation 4.5), according the motivations given in section 4.2. A simple MEP with a single saddle point is reported in figures 5.9 and 5.10, where the energy-dependent k has been used; the three images are located near the saddle point, where the energy increases, giving a better description of the barrier.

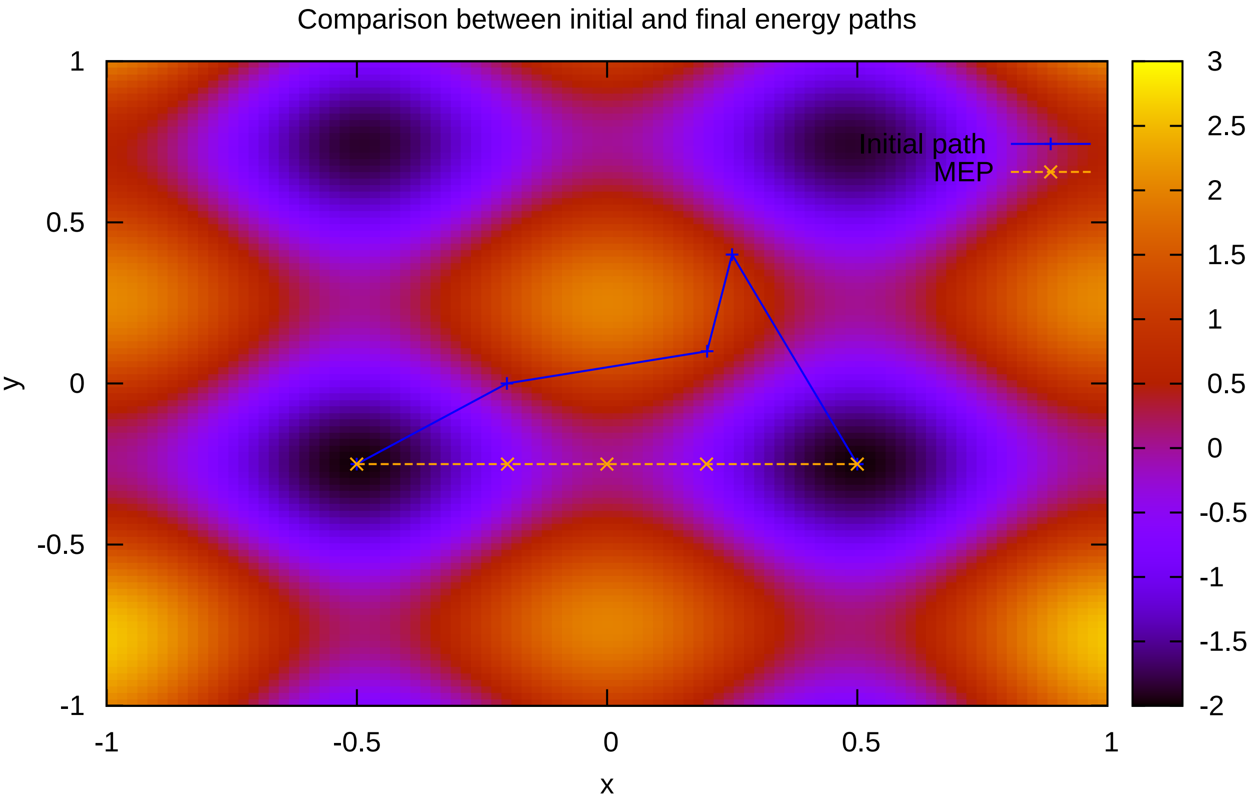


Figure 5.9: Use of energy-dependent k (equation 4.5) for a MEP with a single saddle point. Larger density of points close to it is shown.

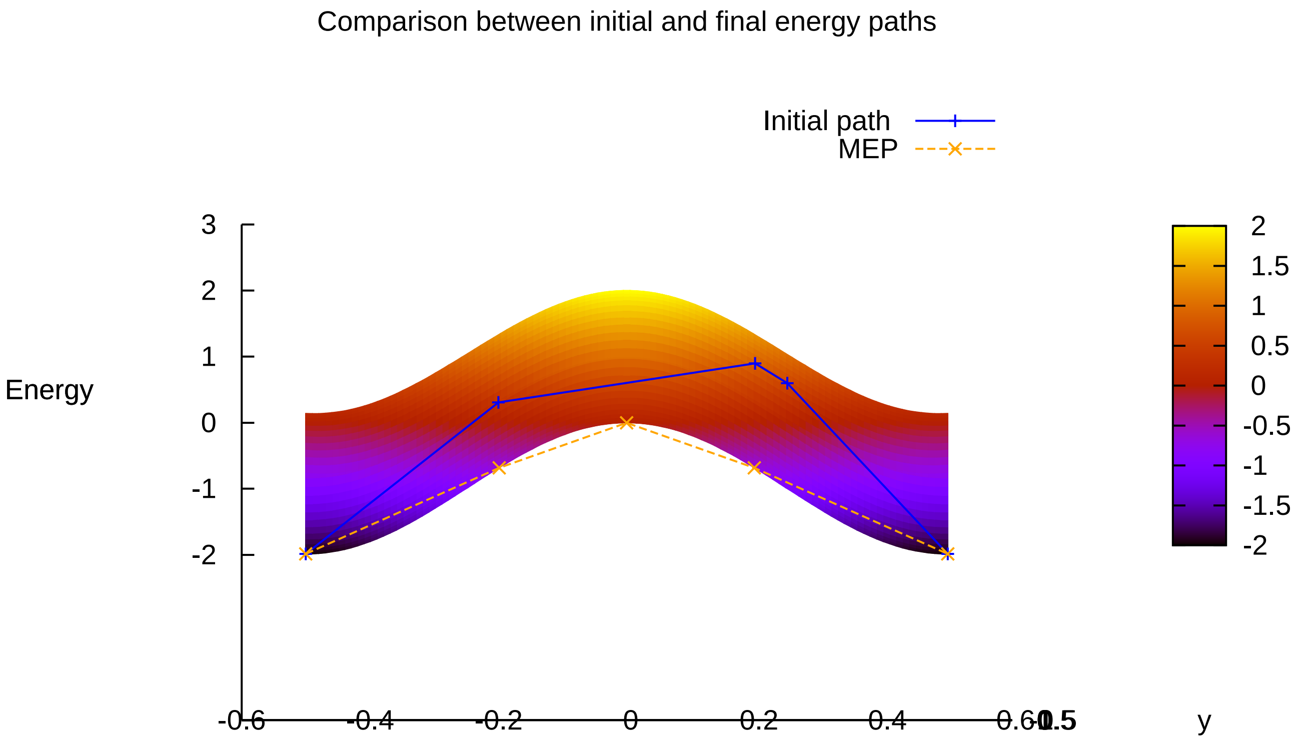


Figure 5.10: Potential energy profile with a initial distorted guess path in blue and the final MEP in orange. Same data of figure 5.9.

5.2 Results with noise

Once the computational machinery has been validated on some test cases, we are now ready to introduce a stochastic perturbation to the minimization procedure on the model potential in order to mimic the behaviour of stochastic minimizers (for instance, Quantum Monte Carlo methods).

White noise

Very interesting results are those obtained inducing white noise on the computations of the true forces (gradient of the energy function). The white noise can be defined as a random function of t whose mean is zero and it is independent at different values of t. Mathematically, we say

$$W(t) \text{ is white noise if } \begin{cases} E\left[W\left(t\right)\right]=0 \quad \forall t \\ cov\left(W\left(t\right),W\left(s\right)\right)=E\left[W\left(t\right)W\left(s\right)\right]=\delta\left(t-s\right) \quad \forall t,s \shortparallel t\neq s \end{cases}$$

and for simulating in the minimization code we have used a random numbers software generator based on a normal distribution with zero mean and unit variance.

Our aim is to study the effect of the noise on the convergence features of the NEB. Considering the tests performed by applying the white noise (error) to the true force of our model energy function

$$\vec{F}_{noise}=\begin{pmatrix} -\dfrac{\partial V(x,y)}{\partial x} \\ \\ -\dfrac{\partial V(x,y)}{\partial y} \end{pmatrix}+s\times\vec{\eta} \tag{5.2}$$

where $\vec{\eta}$ is the random number vector for the noise and $s=\dfrac{\beta}{F_{max}}$ the normalized scaling factor, we have also obtained successful results, what means convergence to the MEP in cases of reasonable values for the noise, as is presented in table 5.3. The MEP resulting is the same as in the potential not affected by noise.

NEB iterations		
s	$k = 0.1$	$k = 10$
15	∞	∞
14	723	∞
13	141	2129
12	202	2730
11	351	1494
10	177	410
7	48	410
5	48	99
3	47	99
1	26	47
0.1	32	46
0.01	30	45
0.001	30	45
No noise	30	45

Table 5.3: Number of NEB iterations for the case of true forces affected by noise, considering the same spring constant $k = 0.1$ and $k = 10$, and varying the scaling factor, s, of white noise applied. In the concrete case of our model energy function and the domain $[-1, 1] \times [-1, 1]$, $F_{max} \simeq 8.3$. We observe convergence even for s values of 10 and bigger, while in QMC methods this value is about 10^{-2}.

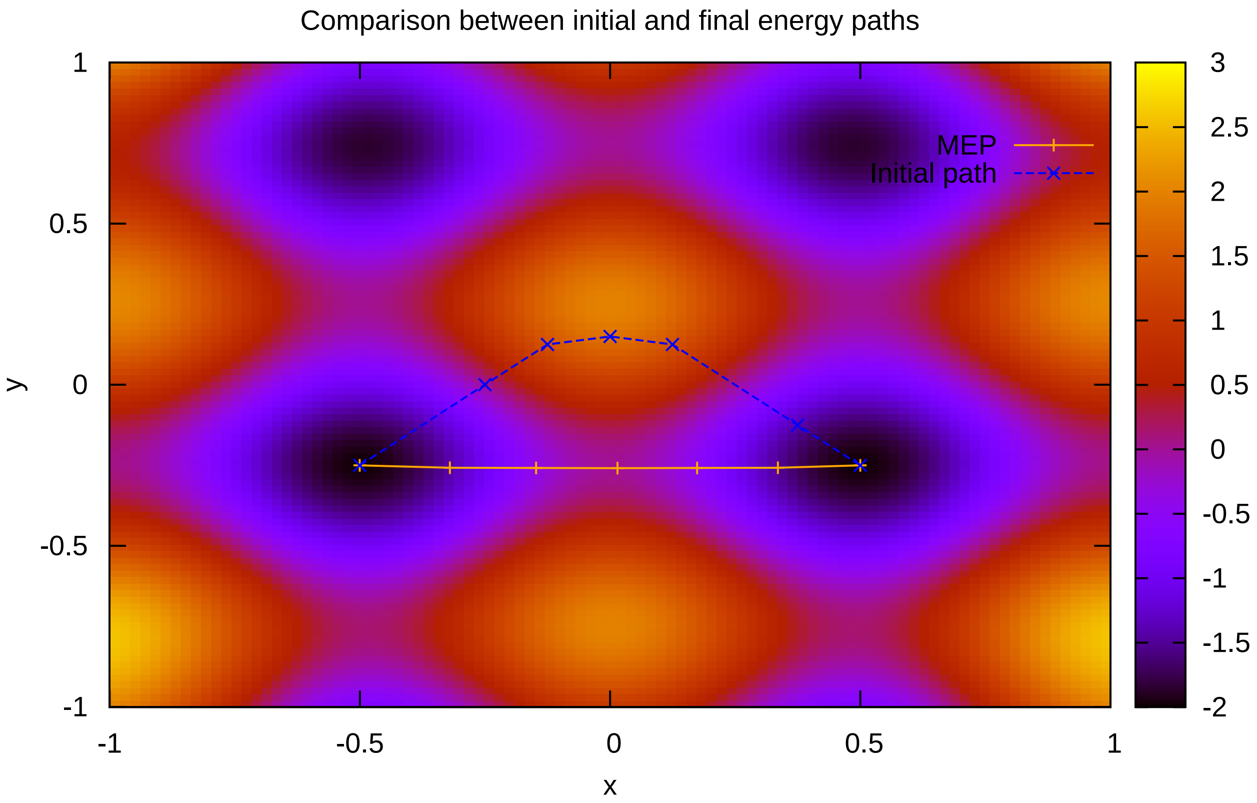


Figure 5.11: Converged MEP with a force affected by random noise. Full convergence is achieved.

Chapter 6

Conclusions

In the theoretical study of chemical reactions, the most challenging quantities to estimate are the energy barriers between minima, which corresponds to the energy of the transition states, located in a saddle point of the potential energy surfaces. There exist several methods to face this problem and the NEB method is one of the most popular due to the low computational cost, the convergence to the MEP instead of only the location of the saddle point as final output, and the fact that it is well described in literature.

We have successfully implemented the NEB algorithm using a general and highly technological framework based on inter-process computer communications. The implementation has been organized according to a server/client model using a socket system, and it has been build up in such a general way that it has the capability to be easily interfaced with all molecular modelling codes.

We have successfully performed tests in a model energy surface considering both simple and multiple transition state pathways. As a second step we have applied our NEB software to potential energy surfaces affected by stochastic errors in order to investigate the capability of the algorithm to be used together with QMC methods. Our results have demonstrated the robustness of the algorithm also in this case, using values of the signal/noise ratio that are well above the ones typically found in QMC calculations. These preliminary results are an encouraging step towards the calculation of reaction pathways by QMC methods.

Bibliography

[1] Andrew R. Leach. *Molecullar Modeling Principles and Applications*. Second. Pearson Education Limited, 2001.

[2] Frank Jensen. *Introduction to Computational Chemistry*. Wiley, 1999.

[3] E. Coccia et al. “Molecular Electrical Properties from Quantum Monte Carlo Calculations: Application to Ethyne”. In: *J. Chem. Theory Comput.* 8 (2012), p. 1952.

[4] B. L. Hammond, W. A. Lester Jr., and P. J. Reynolds. *Monte Carlo Methods in Ab-Initio Quantum Chemistry*. World Scientific, 1994.

[5] W. M. C. Foulkes et al. “Quantum Monte Carlo simulations of solids”. In: *Rev. Mod. Phys.* 73 (2001), p. 33.

[6] D. M. Ceperley. “Path integrals in the theory of condensed helium”. In: *Rev. Mod. Phys.* 67 (1995), p. 279.

[7] S. Baroni and S. Moroni. “Reptation Quantum Monte Carlo: A Method for Unbiased Ground-State Averages and Imaginary-Time Correlations”. In: *”Phys. Rev. Lett.”* 82 (1999), p. 4745.

[8] D. M. Towler. *in ’Computational Methods for Large Systems’*. Wiley, 2011, p. 119.

[9] M. Barborini, S. Sorella, and L. Guidoni. “Structural Optimization by Quantum Monte Carlo: Investigating the Low-Lying Excited States of Ethylene”. In: *J. Chem. Theory Comput.* 8 (2012), p. 1260.

[10] J. Kolorenc and L. Mitas. “Applications of quantum Monte Carlo methods in condensed systems”. In: *Rep. Prog. Phys.* 74 (2011), p. 026502.

[11] Leonardo Spanu, Sandro Sorella, and Giulia Galli. "Nature and Strength of Interlayer Binding in Graphite". In: *Phys. Rev. Lett.* 103 (2009), p. 196401.

[12] R. Maezono et al. "Diamond to beta-tin phase transition in Si within diffusion quantum Monte Carlo". In: *Phys. Rev. B* 82 (2010), p. 184108.

[13] F. Sterpone et al. "Dissecting the Hydrogen Bond: A Quantum Monte Carlo Approach". In: *J. Chem. Theory Comput.* 4 (2008), p. 1428.

[14] S. Sorella, M. Casula, and D. Rocca. "Weak binding between two aromatic rings: Feeling the van der Waals attraction by quantum Monte Carlo methods". In: *J. Chem. Phys.* 127 (2007), p. 14105.

[15] F. Schautz and C. Filippi. "Optimized Jastrow-Slater wave functions for ground and excited states: Application to the lowest states of ethene". In: *J. Chem. Phys.* 120 (2004), p. 10931.

[16] P. M. Zimmermann et al. "Excited states of metylene from Quantum Monte Carlo". In: *J. Chem. Phys.* 131 (2009), p. 124103.

[17] E.G. Lewars. *Computational Chemistry.* Kluwer Academic Publishers, 2003.

[18] B.P. Uberuaga G. Henkelman and H. Jónsson. "A climbing image nudged elastic band method for finding saddle points and minimum energy paths". In: *J. Chem. Phys.* 113 (2000), p. 9901.

[19] G. Jóhannesson G. Henkelman and H. Jónsson. "Methods for finding saddle points and Minimum Energy Path". In: *Theoretical Methods in Condensed Phase Chemistry.* Schwartz, S.D., 2001.

[20] G. Mills H. Jónsson and K.W. Jacobsen. "Nudged elastic band method for finding minimum energy paths of transition". In: *Classical and Quantum Dynamics in Condensed Phase Simulations.* B.J. Berne, G. Ciccotti, and D.F. Coker, 1997. Chap. 16, pp. 385–404.

[21] Janice A. Steckel David S. Sholl. *Density Functional Theory: A Practical Introduction.* Wiley, 2009.

[22] R. Terrel D. Sheppard and G. Henkelman. “Optimization methods for finding minimum energy paths”. In: *J. Chem. Phys.* 128 (2008), p. 134106.

[23] Graeme Henkelman and Hannes Jónsson. “Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points”. In: *J. Chem. Phys.* 113 (2000), p. 9978.